



TUGAS AKHIR - TF 141581

**OPTIMISASI *FOULING RESISTANCE*
PADA *SHELL AND TUBE HEAT EXCHANGER*
MENGUNAKAN *POLLEY MODEL* DAN *RAIN*
*WATER OPTIMIZATION***

ANGGI MALWINDASARI
NRP 2413 100 028

Dosen Pembimbing
Totok Ruki Biyanto, Ph.D.

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TF 141581

***OPTIMIZATION OF SHELL AND TUBE HEAT
EXCHANGER FOULING RESISTANCE THROUGH
GEOMETRIES BY UTILIZING POLLEY MODEL
AND RAIN WATER OPTIMIZATION***

ANGGI MALWINDASARI
NRP 2413 100 028

Supervisor
Totok Ruki Biyanto, Ph.D.

*DEPARTMENT OF ENGINEERING PHYSICS
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017*

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Anggi Malwindasari

NRP : 2413100028

Departemen : Teknik Fisika FTI-ITS

dengan ini menyatakan bahwa tugas akhir saya berjudul Optimisasi *Fouling Resistance* pada *Shell and Tube Heat Exchanger* Menggunakan *Polley Model* dan *Rain Water Optimization* adalah bebas dari plagiasi. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 10 Juli 2017

Yang membuat pernyataan,



Anggi Malwindasari

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

OPTIMISASI *FOULING RESISTANCE* PADA *SHELL AND TUBE HEAT EXCHANGER* MENGUNAKAN *POLLEY MODEL* DAN *RAIN WATER OPTIMIZATION*

TUGAS AKHIR

Oleh :

Anggi Malwindasari

NRP : 2413 100 028

Surabaya, 24 Juli 2017

Mengetahui

Dosen Pembimbing



Totok Ruki Bivanto, Ph.D.

NIPN. 19710702 199802 1 001

Menyetujui,

Ketua Jurusan Teknik Fisika FT-ITS



Agus Muhammad Hatta, S.T., M.Si., Ph.D.

NIPN. 19780902 200312 1 002

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

OPTIMISASI *FOULING RESISTANCE* PADA *SHELL AND TUBE HEAT EXCHANGER* MENGUNAKAN *POLLEY MODEL* DAN *RAIN WATER OPTIMIZATION*





TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Rekayasa Instrumentasi dan Kontrol
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

ANGGI MALWINDASARI
NRP. 2413 100 028

Disetujui oleh Tim Penguji Tugas Akhir :

1. Totok Ruki Biyanto, Ph.D..... (Pembimbing 1)
2. Dr. Katherin Indriawati, S.T., M.T..... (Ketua Penguji)
3. Hendra Cordova, S.T., M.T..... (Penguji 1)
4. Bagus Tris Atmaja, S.T., M.T..... (Penguji 2)

SURABAYA
JULI, 2017

Halaman ini sengaja dikosongkan

**OPTIMISASI *FOULING RESISTANCE*
PADA *SHELL AND TUBE HEAT EXCHANGER*
MENGGUNAKAN POLLEY MODEL DAN
*RAIN WATER OPTIMIZATION***

Nama Mahasiswa : Anggi Malwindasari
NRP : 2413 100 028
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : Totok Ruki Biyanto, Ph.D.

Abstrak

Proses pengolahan energi di industri minyak dan gas membutuhkan unit operasi yang efisien. Salah satu cara pengolahan energi dengan memanfaatkan fenomena perpindahan panas. Hal ini dilakukan sebagai upaya untuk meminimalisasi penggunaan energi. Proses perpindahan panas membutuhkan suatu alat penukar panas. *Heat exchanger* adalah suatu alat termal yang digunakan dalam proses perpindahan panas dua fluida atau lebih yang memiliki temperatur berbeda tanpa terjadi perpindahan massa di dalamnya. Terdapat beberapa tipe *heat exchanger*, salah satu yang sering digunakan adalah tipe *shell* dan *tube*. Permasalahan pada *heat exchanger* yakni terbentuknya *fouling*. *Fouling* adalah deposit yang tidak diinginkan pada permukaan *heat transfer* yang menghambat proses perpindahan panas *heat exchanger* dan meningkatkan hambatan aliran fluida sehingga *pressure drop* menjadi lebih tinggi. *Fouling* merupakan masalah yang tidak dapat dihindari, tetapi dapat diminimalkan dengan menentukan geometri *heat exchanger* yang optimal. Dalam penelitian ini, optimisasi dilakukan untuk menentukan geometri *shell and tube heat exchanger* sehingga diperoleh *fouling resistance* minimal. Optimisasi membutuhkan tiga komponen yakni pemodelan *fouling resistance* dan *heat exchanger*, formulasi fungsi tujuan, dan teknik optimisasi. *Fouling resistance* dimodelkan dengan menggunakan Polley model. Fungsi tujuan pada optimisasi ini adalah meminimalkan *fouling resistance* dengan menentukan geometri *shell and tube heat exchanger*. Batasan optimisasi adalah *heat duty* dan *pressure drop*. Karena

pemodelan *shell and tube heat exchanger* pada optimisasi ini adalah non linier yang memiliki beberapa lokal optimum, maka teknik optimisasi membutuhkan algoritma yang dapat menemukan global optimum. Optimisasi dilakukan dengan menggunakan metode stokastik yakni *Rain Water Optimization*. Variabel yang dioptimisasi adalah diameter dalam *shell* (D_s), diameter luar *tube* (d_o), dan jumlah *baffle* (N_b). Dari hasil optimisasi didapatkan *fouling resistance* yang lebih rendah yakni pada sisi *shell* berkurang 18 % dan di sisi *tube* berkurang 44 % dengan nilai *heat duty* (Q) dan *pressure drop* (Δp) tidak melebihi data desain.

Kata Kunci: optimisasi, *heat exchanger*, *fouling resistance*, geometri.

**OPTIMIZATION OF SHELL AND TUBE HEAT
EXCHANGER FOULING RESISTANCE THROUGH
GEOMETRIES BY UTILIZING POLLEY MODEL AND
RAIN WATER OPTIMIZATION**

Name : Anggi Malwindasari
NRP : 2413 100 028
Department : Engineering Physics FTI-ITS
Supervisor : Totok Ruki Biyanto, Ph.D.

Abstract

Oil and gas industrial process requires efficient operation unit that provides minimum mass and heat waste. Heat waste minimization can be obtained through heat integration by utilizing heat exchanger network. Heat exchanger network consist of heat exchanger that connected in series parallel connection. Heat exchanger as one of thermal operation unit transfer heat between two or more fluids through convection and conduction phenomenon. There are some types of heat exchanger available in the market such as shell and tube heat exchanger. Fouling is deposit accumulation on thermal surface formed by sedimentation, crystallization, corrosion and reaction. Fouling reduce heat exchanger efficiency due to increase thermal resistance and pressure drop. Fouling can not be avoided, however it can be mitigated. One of fouling mitigation is designed heat exchanger optimally. In this research, optimization of heat exchanger design was performed by determining shell and tube heat exchanger geometries. In this optimization require three components i.e. fouling resistance and heat exchanger model, problem formulation and optimization technique. Fouling resistance model was built by utilizing Polley model. The objective of optimization is minimized fouling resistance by determining shell and tube heat exchanger geometries, with constraints available heat duty and allowable pressure drops. Since the optimization fall into mixed non linier problem that have some local optimum, the optimization technique that has capability to find out the global optimum is required. In

this research optimization technique utilize Rain Water Optimization. The selected optimization variables are shellside diameter (D_s), tube outside diameter (d_o) and number of baffle (N_b). Optimization results show shellside fouling resistance was reduced 18%, tubeside was reduced 44% compared with existing design by considering available heat duty and allowable pressure drops.

Keywords: *optimization, heat exchanger, fouling resistance, geometries.*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena rahmat dan hikmat-Nya sehingga penulis diberikan kesehatan, kemudahan, dan kelancaran dalam menyusun laporan tugas akhir yang berjudul:

“OPTIMISASI FOULING RESISTANCE PADA SHELL AND TUBE HEAT EXCHANGER MENGGUNAKAN POLLEY MODEL DAN RAIN WATER OPTIMIZATION”

Tugas akhir ini merupakan salah satu persyaratan akademik yang harus dipenuhi dalam Program Studi S-1 Teknik Fisika FTI-ITS. Penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Totok Ruki Biyanto, Ph.D. selaku dosen pembimbing tugas akhir ini, yang selalu memberikan bimbingan dan semangat pada penulis.
2. Agus M. Hatta, S.T., M.Si., Ph.D. selaku ketua Departemen Teknik Fisika - ITS.
3. Segenap Bapak/Ibu dosen pengajar di Departemen Teknik Fisika - ITS.
4. Sumali dan Winartin selaku orang tua penulis serta segenap keluarga besar penulis yang selalu memberikan doa dan dukungan untuk penulis.
5. Teman-teman Departemen Teknik Fisika – ITS dan teman-teman seperjuangan TA yang senantiasa memberikan motivasi dan bantuan dalam penyelesaian laporan tugas akhir ini.

Penulis menyadari bahwa mungkin masih ada kekurangan dalam laporan ini, sehingga kritik dan saran penulis terima. Semoga laporan ini dapat berguna dan bermanfaat bagi penulis dan pihak yang membacanya.

Surabaya, 6 Juni 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN BEBAS PLAGIASI	v
LEMBAR PENGESAHAN	vii
ABSTRAK	xi
<i>ABSTRACT</i>	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR NOTASI	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Lingkup Kajian	3
BAB II TINJAUAN PUSTAKA	5
2.1 <i>Heat Exchanger</i>	5
2.2 <i>Shell and Tube Heat Exchanger</i>	6
2.3 Konstruksi <i>Shell and Tube Heat Exchanger</i>	7
2.4 Pemodelan <i>Heat Exchanger</i>	9
2.5 <i>Fouling</i>	13
2.6 Pemodelan <i>Fouling Resistance</i>	14
2.7 <i>Rain Water Optimization</i>	15
BAB III METODOLOGI PENELITIAN	17
3.1 Pengambilan Data <i>Heat Exchanger</i>	18
3.2 Pemodelan <i>Heat Exchanger</i>	18
3.3 Pemodelan <i>Fouling Resistance</i>	33
3.4 Validasi Pemodelan <i>Heat Exchanger</i> dan <i>Fouling Resistance</i>	33

3.5 Formulasi Fungsi Tujuan.....	34
3.6 Optimisasi <i>Fouling Resistance</i> pada <i>Shell and Tube Heat Exchanger</i>	34
BAB IV HASIL DAN PEMBAHASAN.....	35
4.1 Hasil Pemodelan <i>Heat Exchanger</i> dan <i>Fouling Resistance</i>	35
4.2 Perancangan dan Pengujian Optimisasi.....	38
4.2.1 Penentuan Fungsi Tujuan	38
4.2.2 Penentuan Variabel yang Dioptimisasi.....	38
4.2.3 Penentuan <i>Properties</i> pada Teknik Optimisasi	41
4.2.4 Optimisasi <i>Fouling Resistance</i> pada <i>Shell and Tube Heat Exchanger</i>	42
4.2.5 Optimisasi <i>Fouling Resistance</i> pada <i>Shell and Tube Heat Exchanger</i> Menggunakan Beberapa Metode Stokastik	45
BAB V PENUTUP.....	49
5.1 Kesimpulan.....	49
5.2 Saran	50
DAFTAR PUSTAKA.....	51
LAMPIRAN	53
BIODATA PENULIS	

DAFTAR GAMBAR

Gambar 2.1 Aliran fluida pada <i>heat exchanger</i>	6
Gambar 2.2 Skema <i>shell and tube heat exchanger</i>	7
Gambar 2.3 Komponen utama <i>shell and tube heat exchanger</i>	8
Gambar 2.4 Tipe <i>shell and tube heat exchanger</i>	9
Gambar 2.5 <i>Fouling</i> pada permukaan <i>heat transfer</i>	144
Gambar 2.6 Diagram alir <i>Rain Water Optimization</i>	166
Gambar 3.1 Diagram alir penelitian.....	17
Gambar 3.2 Geometri <i>basic segmental baffle</i>	20
Gambar 3.3 Panjang <i>tube</i> pada <i>shell and tube heat exchanger</i> ..	20
Gambar 3.4 <i>Tube layout basic parameters</i>	23
Gambar 4.1 Pengaruh diameter dalam <i>shell</i> terhadap <i>fouling</i> <i>resistance</i>	39
Gambar 4.2 Pengaruh diameter luar <i>tube</i> terhadap <i>fouling</i> <i>resistance</i>	40
Gambar 4.3 Pengaruh jumlah <i>baffle</i> terhadap <i>fouling</i> <i>resistance</i>	41
Gambar 4.4 Grafik hasil optimisasi pada <i>software</i> MATLAB ..	43

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 4.1 <i>Properties</i> fluida pada <i>shell and tube heat exchanger</i>	36
Tabel 4.2 Hasil pemodelan <i>heat exchanger</i>	37
Tabel 4.3 Perbandingan geometri <i>heat exchanger</i> dan <i>fouling resistance</i> sebelum dan setelah dioptimisasi.....	44
Tabel 4.4 Perbandingan hasil optimisasi <i>fouling resistance</i> menggunakan beberapa metode stokastik.....	47

Halaman ini sengaja dikosongkan

DAFTAR NOTASI

Variabel	Satuan	Keterangan
d_o	m	Diameter luar <i>tube</i>
n_b		Jumlah <i>baffle</i>
d_s	m	Diameter dalam <i>shell</i>
l_{bi}	m	Jarak <i>baffle</i> inlet
l_{bo}	m	Jarak <i>baffle</i> outlet
L_{bb}	m	Jarak antara <i>bundle</i> dan <i>shell</i>
D_{out}	m	Diameter <i>tube bank outer</i>
L_{ti}	m	Panjang <i>tube</i> efektif
θ_{ds}	rad	Sudut kemiringan pada <i>baffle cut</i>
D_{ctl}	m	Diameter <i>bundle</i>
θ_{ctl}	rad	Sudut kemiringan bagian atas pada <i>baffle cut</i>
L_{bc}	m	Jarak antar <i>baffle</i>
l_{tp}	m	Jarak antar <i>tube</i>
S_m	m ²	Luas aliran <i>crossflow</i> pada sisi <i>shell</i>
F_w		Nilai fraksi <i>tube</i> pada jendela <i>baffle</i>
F_c		Nilai fraksi <i>tube</i> pada <i>pure crossflow</i>
N_{tcc}		Jumlah efektif <i>tube rows</i> pada <i>crossflow</i>
L_{pl}	m	Lebar <i>tube</i> dan partisi diantara <i>tube wall</i>
S_b	m ²	Luas <i>bundle</i> ke <i>shell</i>
F_{sbp}		Perbandingan antara luas area <i>by pass</i> dengan luas <i>crossflow</i> keseluruhan
L_{sb}	m	Jarak antara diameter <i>shell</i> dengan diameter <i>baffle</i>
S_{sb}	m ²	Luas kebocoran <i>shell</i> ke <i>baffle</i>
S_{tb}	m ²	Luas kebocoran <i>tube</i> ke <i>baffle</i>

Variabel	Satuan	Keterangan
J_c		Faktor koreksi pada <i>baffle cut</i> dan jarak antar <i>baffle</i>
r_s		Parameter korelasi
r_{lm}		Parameter korelasi
J_l		Faktor koreksi akibat efek kebocoran <i>baffle</i> antara <i>shell</i> ke <i>baffle</i> dan <i>tube</i> ke <i>baffle</i>
G_s	Kg / s m ²	Kecepatan aliran massa pada <i>shell</i>
Re_s		Bilangan <i>Reynold</i> pada <i>shell</i>
J_b		Faktor koreksi pada <i>bundle by pass flow</i>
J_r		Faktor koreksi yang merugikan temperatur gradien pada aliran laminar
L_i^*		Perbandingan jarak <i>baffle</i> inlet dengan jarak <i>baffle</i> dalam <i>shell</i>
L_o^*		Perbandingan jarak <i>baffle</i> outlet dengan jarak <i>baffle</i> dalam <i>shell</i>
J_s		Faktor koreksi untuk variabel jarak <i>baffle</i> inlet dan outlet
Pr_s		Nilai <i>Prandtl</i> pada <i>shell</i>
j_i		Parameter ideal <i>Colburn</i>
h_i	W/m ² °C	Koefisien perpindahan panas ideal pada sisi <i>shell</i>
h_s	W/m ² °C	Koefisien perpindahan panas pada <i>shell</i>
d_i	m	Diameter dalam <i>tube</i>
A_t	m ²	Luas permukaan <i>tube</i>
G_t	Kg / s m ²	Kecepatan aliran massa pada <i>tube</i>
Re_t		Bilangan <i>Reynold</i> pada <i>tube</i>
Pr_t		Bilangan <i>Prandtl</i> pada <i>tube</i>
h_t	W / m ² °C	Koefisien perpindahan panas pada <i>tube</i>

Variabel	Satuan	Keterangan
α	$\text{m}^2 \text{ K} / \text{J}$	Konstanta deposit
E_a	KJ / mol	Energi aktivasi
γ	$\text{m}^2 \text{ K} / \text{J}$	Konstanta supresi
R	$\text{kJ} / \text{mol K}$	Konstanta gas konstan
R_{fs}	$\text{m}^2 \text{ }^\circ\text{C} / \text{W}$	<i>Fouling resistance</i> pada <i>shell</i>
R_{ft}	$\text{m}^2 \text{ }^\circ\text{C} / \text{W}$	<i>Fouling resistance</i> pada <i>tube</i>
$cond$	$\text{m}^2 \text{ }^\circ\text{C} / \text{W}$	Perpindahan panas konduksi
U_f	$\text{W} / \text{m}^2 \text{ }^\circ\text{C}$	Koefisien perpindahan panas keseluruhan pada kondisi <i>fouling</i>
A_o	m^2	Luas permukaan keseluruhan <i>heat exchanger</i>
N_t		Jumlah <i>tube</i>
F		Faktor koreksi
$LMTD$	$^\circ\text{C}$	<i>Log Mean Temperature Difference</i>
$LMTD_{corr}$	$^\circ\text{C}$	<i>Log Mean Temperature Difference corrected</i>
$T_{c,i}$	$^\circ\text{C}$	Temperatur masuk fluida dingin
$T_{c,o}$	$^\circ\text{C}$	Temperatur keluar fluida dingin
$T_{h,i}$	$^\circ\text{C}$	Temperatur masuk fluida panas
$T_{h,o}$	$^\circ\text{C}$	Temperatur keluar fluida panas
Q	MW	<i>Heat duty</i>
f_s		Faktor friksi pada <i>shell</i>
f_t		Faktor friksi pada <i>tube</i>
Δp_s	psi	<i>Pressure drop</i> pada <i>shell</i>
Δp_t	psi	<i>Pressure drop</i> pada <i>tube</i>

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan industri minyak dan gas di Indonesia yang semakin pesat dari tahun ke tahun menyebabkan kebutuhan energi tiap tahunnya semakin meningkat. Hal ini menyebabkan permintaan kebutuhan energi sebagai sumber daya juga mengalami peningkatan. Proses pengolahan energi salah satunya dengan memanfaatkan fenomena perpindahan panas. Proses tersebut merupakan salah satu upaya penghematan energi agar lebih efisien. Proses perpindahan panas membutuhkan suatu alat penukar panas.

Heat exchanger adalah suatu alat termal yang digunakan untuk proses perpindahan panas yang melibatkan dua atau lebih fluida dengan temperatur yang berbeda. Perpindahan panas pada *heat exchanger* digunakan untuk memindahkan energi dari fluida bertemperatur panas ke fluida bertemperatur dingin ataupun sebaliknya, tanpa terjadi perpindahan massa di dalamnya. Perpindahan panas pada *heat exchanger* dapat terjadi melalui proses konveksi dan konduksi [1].

Alat penukar panas di industri memiliki tipe yang beragam dan dapat diklasifikasikan berdasarkan bentuk *shell* yang didalamnya terdapat banyak *tube* yang disusun dengan jarak tertentu [2]. Salah satu tipe alat penukar panas adalah *shell and tube heat exchanger*. Proses perpindahan panas pada *shell and tube heat exchanger* dipengaruhi oleh geometri *heat exchanger* seperti diameter luar *tube* (d_o), jumlah *baffle* (N_b), dan diameter dalam *shell* (D_s) [3].

Salah satu permasalahan pada *heat exchanger* adalah terbentuknya *fouling*. *Fouling* adalah deposit yang tidak diinginkan pada permukaan *heat transfer* yang menghambat proses perpindahan panas *heat exchanger* dan meningkatkan hambatan aliran fluida sehingga *pressure drop* menjadi lebih tinggi [4]. *Fouling* terjadi ketika fluida berinteraksi dengan material penyusun *heat exchanger*, sehingga beberapa partikel dari fluida akan

mengendap pada permukaan *heat exchanger*. Endapan ini akan terakumulasi dan menurunkan proses perpindahan panas. Hal ini mengakibatkan konsumsi energi pada pompa ataupun kompresor menjadi lebih tinggi. Efisiensi *heat exchanger* semakin menurun dan biaya yang dibutuhkan semakin besar karena perlu dilakukan pembersihan *fouling*.

Beberapa penelitian yang telah dilakukan adalah menganalisa *fouling* akibat kondisi operasi. Yakni *fouling* yang diakibatkan oleh perubahan temperatur dan laju aliran massa. Hingga saat ini *fouling resistance* pada proses desain *heat exchanger* adalah konstanta yang nilainya ditetapkan dahulu kemudian menentukan geometri *heat exchanger*, padahal *fouling resistance* disebabkan karena pemilihan geometri *heat exchanger*. Sehingga *fouling resistance* dan pemilihan geometri *heat exchanger* merupakan hubungan sebab akibat yakni proses desain *heat exchanger* berpengaruh pada *fouling resistance*. Proses desain *heat exchanger* perlu menganalisa *fouling* yang diakibatkan oleh penentuan geometri *heat exchanger*, sehingga dapat ditentukan geometri yang optimal.

Salah satu cara untuk mengurangi *fouling resistance* adalah menentukan geometri yang optimal dari desain *heat exchanger*. Polley model digunakan sebagai pemodelan *fouling resistance* pada *heat exchanger* sehingga dapat ditentukan *fouling resistance* yang minimal [5].

Melihat permasalahan pada *heat exchanger* yakni terjadinya *fouling* yang tidak dapat dihindari tetapi dapat diminimalkan, maka diperlukan penelitian mengenai optimisasi *fouling resistance* pada *shell and tube heat exchanger* menggunakan Polley model dan *Rain Water Optimization*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan yang diambil dalam Tugas Akhir ini antara lain

- a. Bagaimana menentukan geometri *heat exchanger* yang optimal sehingga diperoleh *fouling resistance* yang minimal menggunakan Polley model dan *Rain Water Optimization* ?

- b. Bagaimana *fouling resistance* antara sebelum dan setelah dioptimisasi ?

1.3 Tujuan

Tujuan dilakukannya tugas akhir ini antara lain

- a. Menentukan geometri *heat exchanger* yang optimal sehingga diperoleh *fouling resistance* yang minimal menggunakan Polley model dan *Rain Water Optimization*.
- b. Membandingkan *fouling resistance* antara sebelum dan setelah dioptimisasi.

1.4 Lingkup Kajian

Adapun lingkup kajian pada tugas akhir ini antara lain menggunakan satu buah *heat exchanger* dengan tipe *shell and tube*. Pengambilan data *heat exchanger* meliputi dimensi *heat exchanger*, kondisi operasi dan *properties* fluida. Pemodelan *heat exchanger* meliputi perhitungan koefisien perpindahan panas pada sisi *shell* (h_s) dan *tube* (h_t), koefisien perpindahan panas keseluruhan dalam keadaan *fouling* (U_f), *heat duty* (Q), *pressure drop* (Δp), dan pemodelan *fouling resistance* (R_f) menggunakan Polley model. Nilai yang dioptimisasi adalah *fouling resistance* (R_f). Variabel yang dioptimisasi adalah diameter dalam *shell* (D_s), diameter luar *tube* (d_o), dan jumlah *baffle* (N_b). Teknik optimisasi menggunakan metode stokastik yakni *Rain Water Optimization*.

Halaman ini sengaja dikosongkan

BAB II

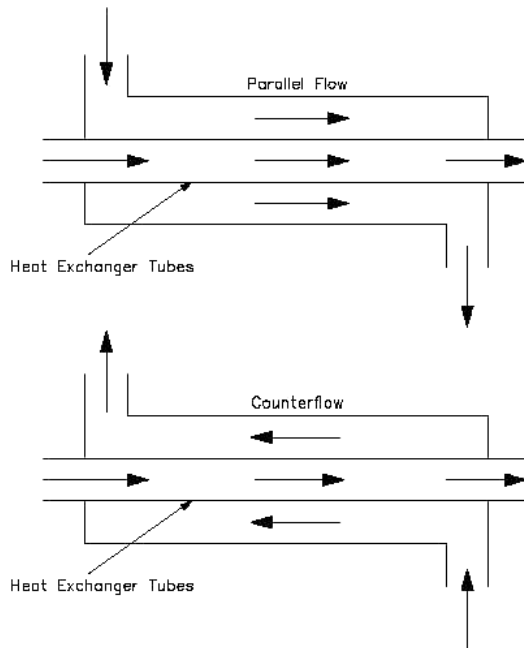
TINJAUAN PUSTAKA

2.1 *Heat Exchanger*

Heat exchanger adalah suatu alat termal yang digunakan dalam proses perpindahan panas dua atau lebih fluida yang memiliki temperatur yang berbeda. Perpindahan panas pada *heat exchanger* digunakan untuk mentransfer energi dari fluida bertemperatur panas ke fluida bertemperatur dingin ataupun sebaliknya, tanpa terjadi perpindahan massa didalamnya. Perpindahan panas pada *heat exchanger* dapat terjadi melalui proses konveksi dan konduksi [1]. Laju perpindahan panas pada *heat exchanger* dipengaruhi oleh koefisien perpindahan panas keseluruhan, luas bidang transfer, dan perbedaan temperatur rata - rata logaritmik [3].

Perpindahan panas juga digunakan untuk mengubah keadaan atau fase suatu fluida, serta menghemat energi pada proses selanjutnya. Fluida-fluida pada *heat exchanger* dipisahkan oleh elemen konduksi berupa pembatas atau dinding berbentuk tabung atau lempengan, sehingga tidak terjadi kontak langsung atau pencampuran fluida-fluida tersebut [3].

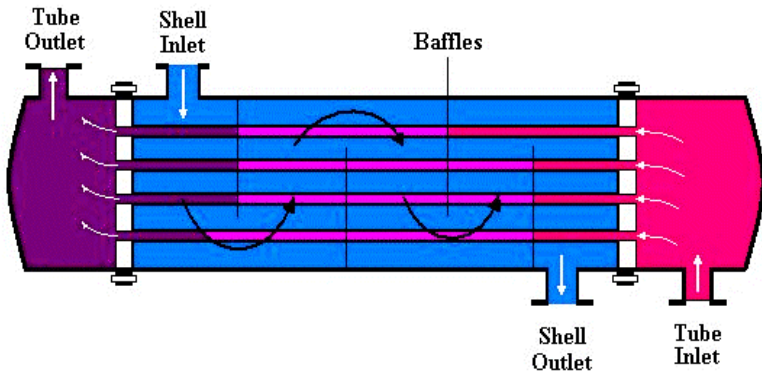
Pada *heat exchanger* fluida panas dan dingin dapat mengalir searah (*co-current flow* atau *parallel flow*) dimana kedua fluida masuk pada ujung *heat exchanger* yang sama dan kedua fluida mengalir searah menuju ujung *heat exchanger* yang lain. Apabila fluida panas dan dingin berlawanan arah (*counter current flow*) dimana fluida yang satu masuk pada ujung *heat exchanger*, sedangkan fluida yang lain masuk pada ujung *heat exchanger* yang lain [6].



Gambar 2.1 Aliran fluida pada *heat exchanger* (a) *parallel-flow* (b) *counter current flow* [6]

2.2 *Shell and Tube Heat Exchanger*

Shell and tube heat exchanger adalah salah satu jenis *heat exchanger* yang biasanya digunakan oleh perusahaan industri seperti minyak dan gas karena *heat exchanger* ini dapat bekerja pada kondisi tekanan relatif tinggi (lebih besar dari 30 bar), memiliki permukaan perpindahan panas per satuan volume yang lebih besar dan lebih mudah dalam proses pembersihan [7]. *Shell and tube heat exchanger* terdiri dari sebuah *shell* yang di dalamnya disusun *tube* dengan rangkaian tertentu untuk mendapatkan luas permukaan yang optimal [2]. Fluida mengalir di *shell* maupun *tube* sehingga terjadi perpindahan panas antara fluida dengan dinding *tube* sebagai perantara.



Gambar 2.2 Skema *shell and tube* heat exchanger

2.3 Konstruksi *Shell and Tube* Heat Exchanger

Komponen utama dari *shell and tube* heat exchanger adalah sebagai berikut :

a. *Tube*

Berbentuk pipa silinder sebagai media mengalirnya fluida yang akan ditukar panasnya sekaligus sebagai pemisah antara fluida di dalam *tube* dengan fluida di dalam *shell*. Kumpulan dari beberapa *tube* diikat oleh *tube bundle*.

b. *Baffle*

Sekat yang berfungsi untuk menahan struktur *tube bundle*, mencegah terjadinya getaran pada *tube*, serta mengatur aliran fluida dalam *shell* sehingga diperoleh turbulensi yang tinggi.

c. *Shell*

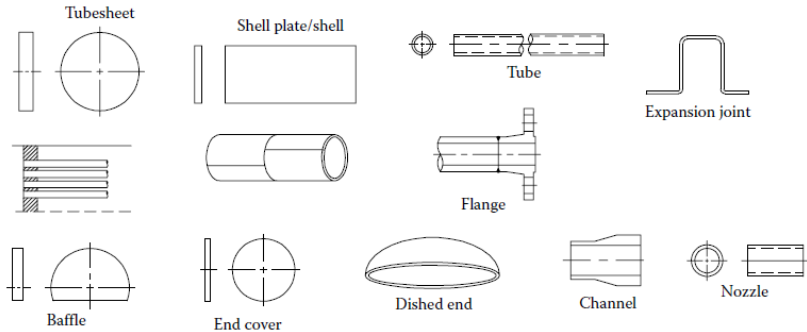
Merupakan selubung yang menyelimuti bagian dalam *heat exchanger* sekaligus sebagai rumah dari *tube bundle*, media mengalirnya fluida yang akan ditukar panasnya serta untuk menahan beban berat, temperatur dan tekanan fluida.

d. *Front head*

Bagian depan *heat exchanger* yang berfungsi tempat mengalirnya fluida.

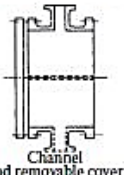
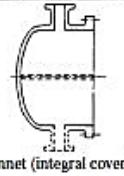
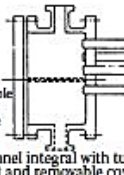
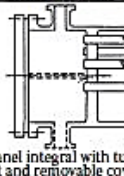

e. *Rear head*

Bagian belakang *heat exchanger* [3].



Gambar 2.3 Komponen utama *shell and tube heat exchanger* [3]

Untuk menentukan dimensi *shell and tube heat exchanger* terdapat standar yang digunakan yakni TEMA. Untuk menentukan tipenya dalam standar TEMA menggunakan kode huruf [8].

Front End Stationary Head Types		Shell Types	Rear End Head Types
A	 Channel and removable cover	E	L
B	 Bonnet (integral cover)	F	M
C	 Removable tube bundle only Channel integral with tube sheet and removable cover	G	N
N	 Channel integral with tube sheet and removable cover	H	P
D	 Special high pressure closure	I	S
		J	T
		K	U
		X	W

Gambar 2.4 Tipe shell and tube heat exchanger berdasarkan standar TEMA [8]

2.4 Pemodelan Heat Exchanger

Heat exchanger banyak digunakan dalam proses industri, spesifikasinya disesuaikan dengan kebutuhan pada proses industri tersebut. Untuk menentukan spesifikasi *heat exchanger* yang

optimal, diperlukan suatu pemodelan. Pemodelan *heat exchanger* meliputi perhitungan koefisien perpindahan panas pada sisi *shell* dan *tube*, koefisien perpindahan panas keseluruhan, *heat duty*, dan *pressure drop*.

2.4.1 Koefisien Perpindahan Panas pada Sisi *Tube* dan *Shell*

Koefisien perpindahan panas pada sisi *tube* dan *shell* berbeda tergantung pada dimensi *heat exchanger*, kondisi operasi serta *properties* fluida.

a. Koefisien perpindahan panas pada sisi *tube*

Koefisien perpindahan panas pada sisi *tube* dapat dihitung dengan menggunakan metode *Bell-Delaware* [3].

$$h_t = \left(\frac{K_t Nu_t}{d_i} \right) \quad (2.1)$$

Dimana,

h_t = koefisien perpindahan panas pada sisi *tube* (W/m² °C)

K_t = konduktifitas termal pada sisi *tube* (W/ m °C)

Nu_t = bilangan *Nusselt* pada fluida di sisi *tube*

d_i = diameter dalam *tube* (m)

Bilangan *Nusselt* untuk aliran yang melewati *tube* dapat dihitung dengan persamaan berikut.

- Untuk aliran turbulen dengan $Re_t > 10^4$ menggunakan persamaan *Side-Tate (Mac Adamas)* [3].

$$Nu_t = 0,027 Re_t^{0,8} Pr_t^{1/3} \left(\frac{\mu_t}{\mu_w} \right)^{0,14} \quad (2.2)$$

- Untuk aliran *intermediate* dengan $2100 < Re_t < 10^4$ menggunakan persamaan *Colburn* [3].

$$Nu_t = 0,023 Re_t^{0,8} Pr_t^{0,4} \left(\frac{\mu_t}{\mu_w} \right)^{0,14} \quad (2.3)$$

- Untuk aliran laminar dengan $Re_t \leq 2100$ menggunakan persamaan *Side-Tate (Mac Adamas)* [3].

$$Nu_t = 1,86(Re_t Pr_t d_i/L)^{0,5} Pr_t^{1/3} \left(\frac{\mu_t}{\mu_w}\right)^{0,14} \quad (2.4)$$

Dimana,

Re_t = bilangan *Reynold* pada sisi *tube*

d_i = diameter dalam *tube* (m)

Pr_t = bilangan *Prandtl* pada sisi *tube*

L = panjang *tube* (m)

μ_t = viskositas fluida pada sisi *tube* (kg/m s)

μ_w = viskositas fluida pada temperatur *wall* sisi *tube* (kg/m s)

- b. Koefisien perpindahan panas pada sisi *shell*

Koefisien perpindahan panas pada sisi *shell* dengan menggunakan metode *Bell-Delaware* [3].

$$h_s = h_i J_c J_b J_s J_r \quad (2.5)$$

Dimana,

h_s = koefisien perpindahan panas pada sisi *shell* (W/m² °C)

h_i = koefisien perpindahan panas ideal pada sisi *shell* (W/m² °C)

J_c = faktor koreksi pada *baffle cut* dan jarak antar *baffle*

J_l = faktor koreksi akibat efek kebocoran *baffle* antara *shell* ke *baffle* dan *tube* ke *baffle*

J_b = faktor koreksi pada *bundle by pass flow*

J_s = faktor koreksi pada variabel jarak *baffle* sisi inlet dan outlet

J_r = faktor koreksi yang merugikan temperatur gradien pada aliran laminar

2.4.2 Laju Perpindahan Panas pada *Heat Exchanger*

Untuk menentukan laju Perpindahan Panas pada *heat exchanger* menggunakan persamaan sebagai berikut [9].

$$Q = U LMTD_{corr.} A_o \quad (2.6)$$

Dimana,

Q	= laju perpindahan panas pada <i>shell and tube heat exchanger</i> (W)
U	= koefisien perpindahan panas keseluruhan (W/m ² °C)
$LMTD_{corr.}$	= <i>Log Mean Temperature Difference corrected</i> (°C)
A_o	= luas permukaan perpindahan panas keseluruhan (m ²)

Log Mean Temperature Difference corrected ($LMTD_{corr.}$) dapat ditentukan dengan menggunakan persamaan berikut.

$$LMTD_{corr.} = LMTD \cdot F \quad (2.7)$$

Dimana,

$LMTD$	= <i>Log Mean Temperature Difference</i> (°C)
F	= faktor koreksi

Log Mean Temperature Difference ($LMTD$) merupakan rata-rata perbedaan temperatur antara fluida panas dan fluida dingin. $LMTD$ dapat ditentukan dengan menggunakan persamaan berikut [6].

$$LMTD = \frac{(T_{h,i} - T_{c,o}) - (T_{h,o} - T_{c,i})}{\ln \left(\frac{T_{h,i} - T_{c,o}}{T_{h,o} - T_{c,i}} \right)} \quad (2.8)$$

Faktor koreksi (F) dapat dihitung dengan menggunakan persamaan berikut.

$$F = \frac{\sqrt{R^2 + 1} \ln(1 - S) / (1 - RS)}{(R - 1) \ln \frac{2 - S(R + 1 - \sqrt{R^2 + 1})}{2 - S(R + 1 + \sqrt{R^2 + 1})}} \quad (2.9)$$

Dengan nilai R didapatkan dari persamaan berikut.

$$R = \frac{T_{h,i} - T_{h,o}}{T_{c,o} - T_{c,i}} \quad (2.10)$$

dan nilai S dari persamaan berikut.

$$S = \frac{T_{c,o} - T_{c,i}}{T_{h,i} - T_{c,i}} \quad (2.11)$$

Dimana,

R, S = faktor efisiensi temperatur

$T_{c,i}$ = temperatur masuk fluida dingin ($^{\circ}\text{C}$)

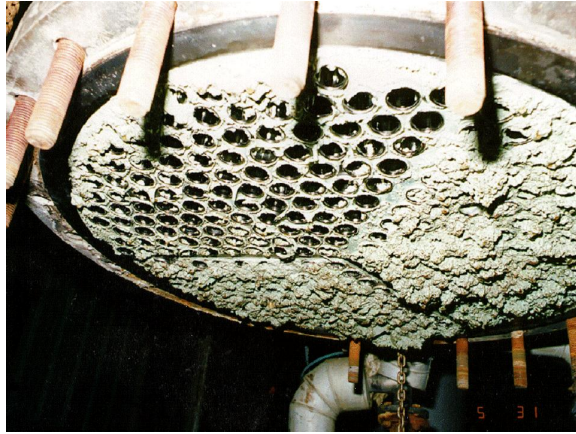
$T_{c,o}$ = temperatur keluar fluida dingin ($^{\circ}\text{C}$)

$T_{h,i}$ = temperatur masuk fluida panas ($^{\circ}\text{C}$)

$T_{h,o}$ = temperatur keluar fluida panas ($^{\circ}\text{C}$)

2.5 Foulings

Fouling adalah deposit yang tidak diinginkan pada permukaan *heat transfer* yang menghambat proses perpindahan panas *heat exchanger* dan meningkatkan hambatan aliran fluida sehingga *pressure drop* menjadi lebih tinggi [4]. *Fouling* terjadi karena proses sedimentasi, kristalisasi, reaksi dan korosi. Dalam proses pengolahan minyak mentah atau *refinery*, penyebab *fouling* terutama disebabkan oleh kotoran yang dibawa oleh minyak mentah. *Fouling* tidak dapat dihindari pada *heat exchanger* dan mempengaruhi produktivitas dan efisiensi *heat exchanger*. Faktor yang mempengaruhi *fouling* dalam industri minyak antara lain suhu permukaan, kecepatan aliran fluida, jenis minyak mentah, campuran minyak mentah, dan reaksi [10].



Gambar 2.5 *Fouling* pada permukaan *heat transfer shell and tube* [11]

2.6 Pemodelan *Fouling Resistance*

Fouling tidak dapat dihindari dalam berbagai proses di industri, oleh karena itu perlu dilakukan minimalisasi *fouling*. Beberapa parameter diperlukan untuk memodelkan *fouling resistance*. Penelitian yang telah dilakukan untuk memodelkan *fouling resistance* diantaranya menggunakan model semi empiris [12]. Pada tahun 2002 Polley memodelkan *fouling resistance* dengan menggunakan variabel bilangan *Reynold* (Re), bilangan *Prandtl* (Pr) dan temperatur *wall* (T_w) dengan menggunakan beberapa parameter yakni α (konstanta deposit), γ (konstanta supresi), R (konstanta gas konstan), dan E_a (energi aktivasi) [13]. Berikut adalah persamaan Polley.

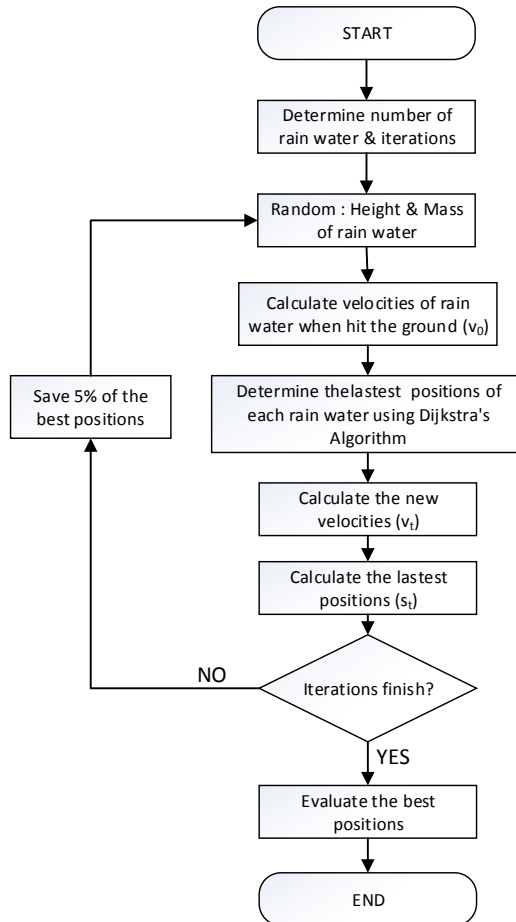
$$\frac{dR_f}{dt} = \alpha Re^{-0,8} Pr^{-\frac{1}{3}} \exp\left(-\frac{E_a}{RT_w}\right) - \gamma Re^{0,8} \quad (2.12)$$

Dimana nilai parameter telah ditetapkan sebagai berikut.

$$\begin{aligned} \alpha &= 277,8 \text{ m}^2\text{K/J} \\ E_a &= 48 \text{ KJ/mol} \\ \gamma &= 4,17 \cdot 10^{-13} \text{ m}^2\text{K/J} \\ R &= 0,008314462 \text{ KJ mol / K} \end{aligned}$$

2.7 *Rain Water Optimization*

Secara umum *Rain Water Optimization* merupakan teknik optimisasi berdasarkan filosofi hujan yang turun. Hujan yang turun tersebut diasumsikan merupakan objek yang jatuh bebas dari langit untuk mencari posisi yang paling rendah di bumi sesuai dengan hukum *Newton* tentang gerak. Proses optimisasi dimulai dengan melakukan inisiasi untuk menentukan parameter seperti jumlah *rain water*, dimensi variabel optimisasi, batas atas dan bawah variabel optimisasi, jumlah iterasi dan global optimum untuk menemukan nilai minimum. Masing-masing *rain water* memiliki ketinggian dan massa yang nilainya ditentukan secara acak. Kecepatan dan posisi tiap-tiap *rain water* dihitung ketika mencapai tanah. *Rain water* akan mengalir ke posisi paling rendah dengan kecepatan tertentu. Posisi baru *rain water* akan diperbarui hingga iterasi selesai [14]. Berikut adalah diagram alir *Rain Water Optimization*.

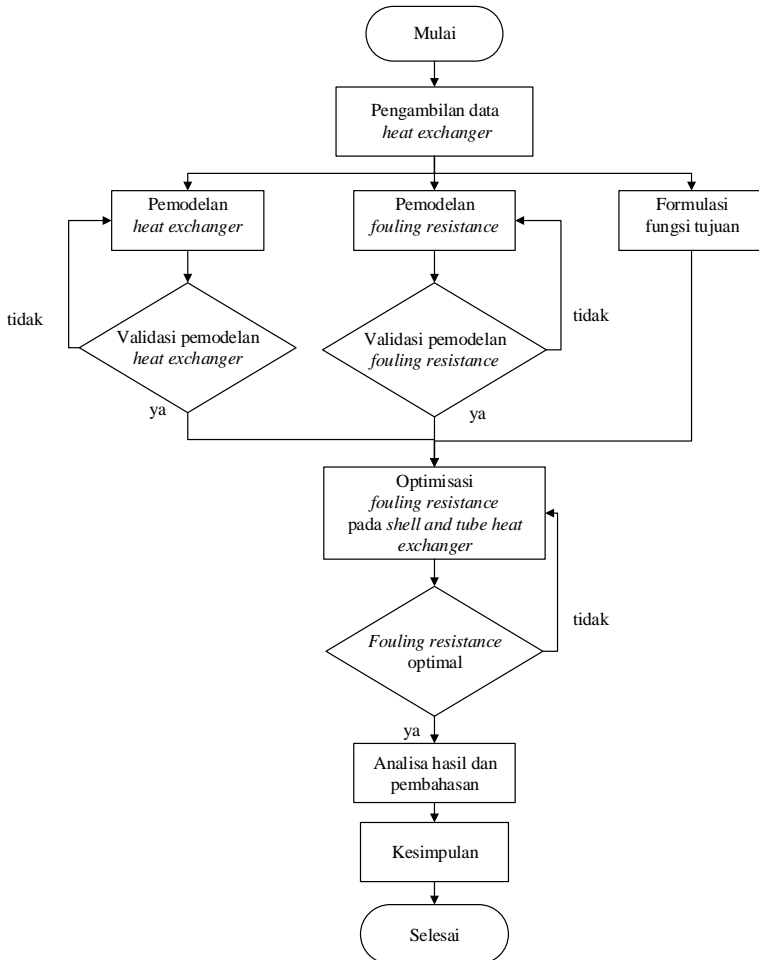


Gambar 2.6 Diagram alir *Rain Water Optimization* [14]

BAB III

METODOLOGI PENELITIAN

Penelitian ini dirancang melalui beberapa tahap yang digambarkan pada diagram alir sebagai berikut.



Gambar 3.1 Diagram alir penelitian

3.1 Pengambilan Data *Heat Exchanger*

Pengambilan data *heat exchanger* dilakukan di PT. Pertamina RU IV- Cilacap. Adapun data yang diambil diantaranya adalah :

- a. Data dimensi *heat exchanger* (seperti terlampir)
- b. Data *properties* fluida pada sisi *shell* dan *tube heat exchanger*, yang meliputi :
 - Massa jenis fluida pada sisi *shell* dan *tube*
 - Kapasitas panas fluida pada sisi *shell* dan *tube*
 - Viskositas fluida pada sisi *shell* dan *tube*
 - Koefisien konduktifitas termal pada sisi *shell* dan *tube*
- c. Data kondisi operasi yang terdiri dari temperatur di inlet dan outlet sisi *shell* dan *tube* dan laju aliran massa di sisi *shell* dan *tube*.

3.2 Pemodelan *Heat Exchanger*

Pemodelan *heat exchanger* digunakan untuk menentukan koefisien perpindahan panas sisi *shell* dan *tube*, koefisien perpindahan panas keseluruhan, *heat duty*, dan *pressure drop*. Data-data yang digunakan untuk memodelkan *heat exchanger* berupa paramater yang sudah ditetapkan nilainya, serta variabel yang nilainya akan diubah-ubah hingga mencapai kondisi yang optimal. Adapun variabel tersebut adalah diameter dalam *shell* (D_s), diameter luar *tube* (d_o) dan jumlah *baffle* (N_b). Sedangkan parameter yang ditetapkan nilainya diantaranya adalah :

- Konduktivitas material pada *tube* (K_{cs})
- Panjang *tube* (L_{ta})
- *Baffle cut* (B_c)
- Jumlah *shell* (N_s)
- Jumlah *tube pass* (N_p)
- *Tube layout* (θ_{tp})
- Jumlah *sealing strips* (N_{ss})
- Laju aliran masa pada *tube* (m_t)
- Viskositas fluida pada *tube* (μ_t)
- Massa jenis fluida pada *tube* (ρ_t)
- Konduktivitas termal pada *tube* (K_t)

- Kalor jenis fluida pada *tube* ($C_{p,i}$)
- Laju aliran masa pada *shell* (m_s)
- Viskositas fluida pada *shell* (μ_s)
- Kalor jenis fluida pada *shell* ($C_{p,s}$)
- Massa jenis aliran pada *shell* (ρ_s)
- Konduktivitas termal pada *shell* (K_s)
- Temperatur masuk fluida dingin ($T_{c,i}$)
- Temperatur keluar fluida dingin ($T_{c,o}$)
- Temperatur masuk fluida panas ($T_{h,i}$)
- Temperatur keluar fluida panas ($T_{h,o}$)

3.2.1 Perhitungan Perpindahan Panas pada Sisi *Shell*

Metode yang digunakan untuk menghitung koefisien perpindahan panas pada sisi *shell* adalah *Bell-Delaware's Method*. Adapun persamaan-persamaan yang digunakan adalah sebagai berikut.

Persamaan untuk menghitung jarak antara *bundle* dan *shell* (L_{bb}).

$$L_{bb} = \frac{(12+0,005 D_s)}{1000} \quad (3.1)$$

Dimana

L_{bb} = jarak antara *bundle* dan *shell* (m)

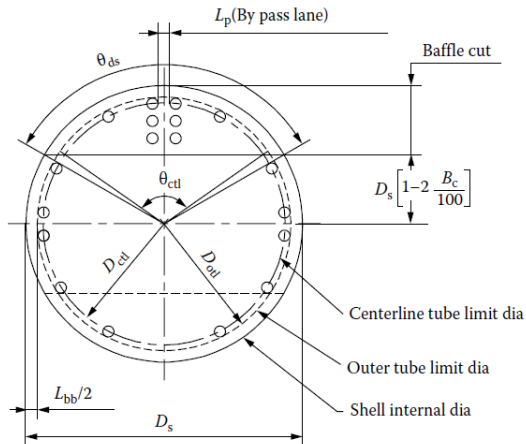
D_s = diameter dalam *shell* (m)

Persamaan untuk menghitung nilai diameter *tube bank outer* (D_{otl}).

$$D_{otl} = D_s - L_{bb} \quad (3.2)$$

Dimana,

D_{otl} = diameter *tube bank outer* (m)

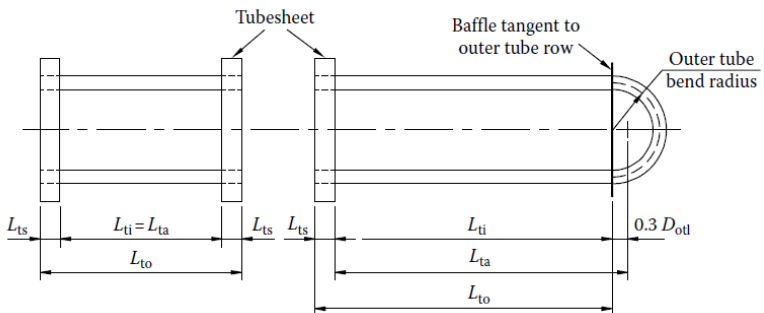


Gambar 3.2 Geometri *basic segmental baffle* [3]

Diameter *bundle* ditentukan dengan persamaan di bawah ini.

$$D_{ctl} = D_{otl} - d_o \quad (3.3)$$

Dimana,

$$D_{ctl} = \text{diameter } bundle \text{ (m)}$$
$$d_o = \text{diameter luar } tube \text{ (m)}$$


Gambar 3.3 Panjang *tube* pada *shell and tube heat exchanger* [3]

Persamaan untuk menghitung jarak antar *baffle* (L_{bc})

$$L_{bc} = \frac{L_{ta}}{N_b + 1} \quad (3.4)$$

Dimana,

N_b = jumlah *baffle*

Persamaan untuk menghitung nilai L_{ti} .

$$L_{ti} = ((N_b - 1)L_{bc}) + L_{bi} + L_{bo} \quad (3.5)$$

Dimana,

N_b = jumlah *baffle*

L_{bc} = jarak antar *baffle* (m)

L_{bi} = jarak *baffle* inlet (m)

L_{bo} = jarak *baffle* outlet (m)

Sudut kemiringan pada *baffle cut* adalah sudut pada pusat perpotongan antara *baffle cut* dan *inner shell*.

θ_{ds} dapat dihitung dengan menggunakan persamaan berikut.

$$\theta_{ds} = 2 \cos^{-1} \left(1 - \frac{2B_c}{100} \right) \quad (3.6)$$

Dimana,

θ_{ds} = Sudut kemiringan pada *baffle cut* (rad)

B_c = *baffle cut* (%)

Persamaan sudut kemiringan bagian atas pada *baffle cut*.

$$\theta_{ctl} = 2 \cos^{-1} \left[\frac{D_s}{D_{ctl}} \left(1 - \frac{2B_c}{100} \right) \right] \quad (3.7)$$

Dimana,

θ_{ctl} = sudut kemiringan bagian atas pada *baffle cut* (rad)

Luas aliran *crossflow* pada sisi *shell* ditentukan dengan persamaan berikut.

$$S_m = L_{bc} \left[L_{bb} + \frac{D_{ctl}}{L_{tp,eff}} (L_{tp} - do) \right] \quad (3.8)$$

Dimana,

S_m = luas aliran *crossflow* pada sisi *shell* (m²)

$L_{tp,eff}$ = jarak efektif antar *tube* (m)

Ketika *tube layout* yang dipilih 30° dan 90° ($\theta_p = 30^\circ$ atau 90°) maka, $L_{tp,eff}$ sama dengan panjang antar *tube* atau *tube pitch* ($L_{tp,eff} = L_{tp}$). Ketika *tube layout* yang digunakan adalah 45° ($\theta_p = 45^\circ$) maka $L_{tp,eff}$ sama dengan 0,707 L_{tp} ($L_{tp,eff} = 0,77 L_{tp}$). Perhitungan sudut kemiringan bagian atas pada *baffle cut* digunakan untuk menentukan nilai fraksi *tube* pada jendela *baffle* dan pada *pure crossflow*.

$$F_w = \frac{\theta_{ctl}}{2\pi} - \frac{\sin \theta_{ctl}}{2\pi} \quad (3.9)$$

Dimana,

F_w = nilai fraksi *tube* pada jendela *baffle*

$$F_c = 1 - 2F_w \quad (3.10)$$

Dimana,

F_c = nilai fraksi *tube* pada *pure crossflow*

Persamaan jumlah efektif *tube rows* pada *crossflow*.

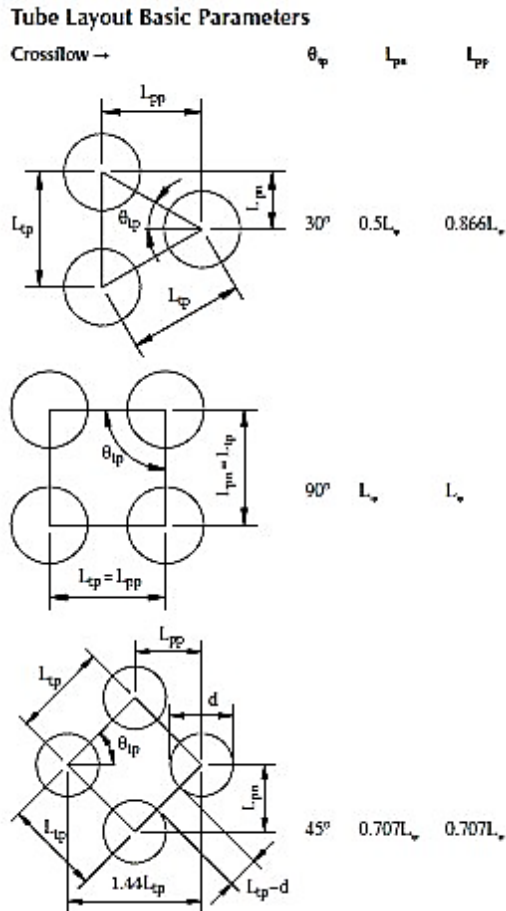
$$N_{tcc} = \frac{D_s}{L_{pp}} \left[1 - \frac{2 B_c}{100} \right] \quad (3.11)$$

Dimana,

N_{tcc} = jumlah efektif *tube rows* pada *crossflow*

L_{pp} = jarak efektif antar *tube rows* (m)

L_{pp} dapat dilihat pada gambar *tube layout basic parameters*.



Gambar 3.4 *Tube layout basic parameters* [3]

Luas *bundle* ke *shell* dapat dihitung dengan persamaan berikut.

$$S_b = L_{bc} (D_s - D_{otl} + L_{pl}) \quad (3.12)$$

Dimana,

S_b = luas *bundle* ke *shell* (m^2)

L_{pl} = lebar *tube* dan partisi diantara *tube wall* (m)

Untuk standar dalam perhitungan, nilai $L_{pl} = 0$

Persamaan rasio antara luas area *by pass* dengan luas *crossflow* keseluruhan adalah

$$F_{sbp} = \frac{S_b}{S_m} \quad (3.13)$$

Dimana,

F_{sbp} = perbandingan antara luas area *by pass* dengan luas *crossflow* keseluruhan

Jarak antara diameter *shell* dengan diameter *baffle* dihitung dengan persamaan berikut.

$$L_{sb} = \frac{3,1 + 0,004 D_s}{1000} \quad (3.14)$$

Dimana,

L_{sb} = Jarak antara diameter *shell* dengan diameter *baffle* (m)

Persamaan luas kebocoran *shell* ke *baffle* dengan *circle segment* pada *baffle* adalah

$$S_{sb} = \pi D_s \frac{L_{sb}}{2} \left(\frac{2\pi - 2\theta_{ds}}{2\pi} \right) \quad (3.15)$$

Dimana,

S_{sb} = luas kebocoran *shell* ke *baffle* (m²)

Persamaan luas kebocoran *tube* ke *baffle* adalah

$$S_{tb} = \frac{\pi}{4} [(d_o + L_{tb})^2 - d_o^2] N_t (1 - F_w) \quad (3.16)$$

Dimana,

S_{tb} = luas kebocoran *tube* ke *baffle* (m²)

L_{tb} = Jarak diameter luar *tube* dengan lubang *baffle* (m)

Standar TEMA merekomendasikan jarak sebagai fungsi diameter *tube* dan jarak *baffle* dengan nilai 0,0008 m atau 0,0004 m.

Untuk menghitung koefisien perpindahan panas pada *shell*, maka perlu menghitung faktor-faktor koreksi pada *shell*.

Faktor koreksi J_c dihitung dengan persamaan berikut.

$$J_c = 0,55 + 0,72F_c \quad (3.17)$$

Dimana,

J_c = Faktor koreksi pada *baffle cut* dan jarak antar *baffle*

Faktor koreksi J_l dengan menggunakan persamaan berikut.

$$J_l = 0,44(1 - r_s) + [1 - 0,44(1 - r_s)]e^{-2,2r_{lm}} \quad (3.18)$$

Dengan nilai r_s dan r_{lm}

$$r_s = \frac{S_{sb}}{S_{sb} + S_{tb}} \quad (3.19)$$

$$r_{lm} = \frac{S_{sb} + S_{tb}}{S_m} \quad (3.20)$$

Dimana,

J_l = faktor koreksi akibat efek kebocoran *baffle* antara *shell* ke *baffle* dan *tube* ke *baffle*

r_s dan r_{lm} = parameter korelasi

Persamaan kecepatan aliran massa dan bilangan *Reynold*.

$$G_s = \frac{m_s}{S_m} \quad (3.21)$$

$$Re_s = \frac{G_s d_o}{\mu_s} \quad (3.22)$$

Dimana,

G_s = kecepatan aliran massa pada *shell* (kg/ s m²)

m_s = aliran massa pada *shell* (kg/s)

μ_s = viskositas fluida pada *shell* (kg/m s)

Re_s = bilangan *Reynold* pada *shell*

Faktor koreksi J_b dihitung dengan persamaan berikut.

$$J_b = \exp\{-C_{bh}F_{sbp}[1 - 2r_s]\} \quad (3.23)$$

Dimana,

J_b = faktor koreksi pada *bundle by pass flow*

C_{bh} = 1,25 pada kondisi aliran laminar ($Re_s \leq 100$)

C_{bh} = 1,35 pada kondisi aliran turbulen ($Re_s \geq 100$)

Faktor koreksi J_r dengan menggunakan persamaan berikut.

$$J_r = \frac{1,51}{N_c^{0,18}} \quad (3.24)$$

$$N_c = (N_{tcc} + N_{tcw})(N_b + 1) \quad (3.25)$$

Dimana,

J_r = faktor koreksi yang merugikan temperatur gradien pada aliran laminar

N_c = jumlah total *tube rows* pada *heat exchanger*

Persamaan di atas digunakan ketika aliran laminar $Re_s < 20$

Ketika Re_s berada pada $20 \leq Re_s \leq 100$ menggunakan persamaan sebagai berikut

$$J_r = \frac{1,51}{N_c^{0,18}} + \left(\frac{20 - Re_s}{80}\right) \left(\frac{1,51}{N_c^{0,18}} - 1\right) \quad (3.26)$$

Dan ketika kondisi aliran pada *shell* turbulen ($Re_s \geq 100$), maka faktor koreksinya bernilai satu ($J_r = 1$).

Faktor koreksi J_s dihitung dengan persamaan berikut.

$$J_s = \frac{(N_b - 1) + (L_i^*)^{1-n} + (L_o^*)^{1-n}}{(N_b - 1) + (L_i^* - 1) + (L_o^* - 1)} \quad (3.27)$$

$$L_i^* = \frac{L_{bi}}{L_{bc}} \quad (3.28)$$

$$L_o^* = \frac{L_{bo}}{L_{bc}} \quad (3.29)$$

Dimana,

J_s = faktor koreksi untuk variabel jarak *baffle* inlet dan outlet

L_i^* = perbandingan jarak *baffle* inlet dengan jarak *baffle* dalam *shell* (m)

L_o^* = perbandingan antara jarak *baffle* outlet dengan jarak *baffle* dalam *shell* (m)

Untuk parameter n ketika alirannya turbulen $n=0,6$ dan ketika alirannya laminar $n=1$

Persamaan bilangan *Prandtl* pada *shell*.

$$Pr_s = \frac{\mu_s C_{ps}}{k_s} \quad (3.30)$$

Dimana,

Pr_s = bilangan *Prandtl*

C_{ps} = kapasitas panas spesifik pada *shell* (J/kg °C)

K_s = konduktifitas termal pada *shell* (W/ m °C)

μ_s = viskositas fluida pada *shell* (kg/m s)

Pada sisi *shell* dapat dihitung nilai perpindahan panas ideal.

$$h_i = \frac{j_i C_{ps} G_s (\Delta T_s)^n}{Pr_s^{\frac{2}{3}}} \quad (3.31)$$

Dengan j_i adalah parameter ideal *Colburn* dihitung dengan persamaan sebagai berikut:

$$j_i = 1,73 Re_s^{-0,694} \text{ ketika } 1 \leq Re_s \leq 100 \quad (3.32)$$

$$j_i = 0,717 Re_s^{-0,574} \text{ ketika } 100 \leq Re_s \leq 1000 \quad (3.33)$$

$$j_i = 0,236 Re_s^{-0,346} \text{ ketika } 1000 \leq Re_s \quad (3.34)$$

Untuk mencari nilai parameter ideal *Colburn* $(\phi_s)^n$.

$$(\phi_s)^n = \left(\frac{\mu_s}{\mu_{sw}} \right)^{0,14} \quad (3.35)$$

Dimana,

h_i = koefisien perpindahan panas ideal pada sisi *shell* (W/ m² °C)

j_i = parameter ideal *Colburn*

μ_{sw} = viskositas fluida pada temperatur *wall* sisi *shell* (kg/m s)

Persamaan nilai koefisien perpindahan panas pada *shell*.

$$h_s = h_i J_c J_b J_s J_r \quad (3.36)$$

Dimana,

h_s = koefisien perpindahan panas pada *shell* (W/ m² °C)

3.2.2 Perhitungan Perpindahan Panas pada Sisi *Tube*

Metode yang digunakan untuk menghitung koefisien perpindahan panas pada sisi *tube* adalah *Bell-Delaware's Method*. Adapun persamaan-persamaan untuk menentukan nilai koefisien perpindahan panas pada *tube* sebagai berikut.

Luas permukaan *tube* dihitung dengan persamaan berikut.

$$A_t = \frac{\pi d_i^2 N_t}{4} \quad (3.37)$$

Dimana diameter dalam *tube* menggunakan persamaan berikut

$$d_i = d_o - 2 t_w \quad (3.38)$$

Jumlah *tube* dalam *shell* menggunakan persamaan berikut

$$N_t = \frac{0,78 d_{ctl}^2}{L_{tp}^2} \quad (3.39)$$

Dimana,

A_t = luas permukaan *tube* (m²)

d_i = diameter dalam *tube* (m)

d_o = diameter luar *tube* (m)

N_t = jumlah *tube*

t_w = ketebalan *tube* (m)

D_{ctl} = diameter *bundle* (m)

L_{tp} = jarak antar *tube* (m)

Kecepatan aliran massa pada *tube* dihitung dengan persamaan berikut.

$$G_t = \frac{m_t N_p}{A_t} \quad (3.40)$$

Dimana,

G_t = kecepatan aliran massa pada *tube* (kg/ s m²)

m_t = aliran massa pada *tube* (kg/s)

N_p = jumlah *tube pass*

Persamaan bilangan *Reynold* adalah

$$Re_t = \frac{G_t d_i}{\mu_t} \quad (3.41)$$

Dimana

Re_t = bilangan *Reynold* pada *tube*

μ_t = viskositas fluida pada *tube* (kg/m s)

Bilangan *Prandtl* pada *tube* dapat dihitung dengan persamaan berikut

$$Pr_t = \frac{c_{p_t} \mu_t}{K_t} \quad (3.42)$$

Dimana,

C_{pt} = kapasitas panas spesifik pada *tube* (J/kg °C)

K_t = konduktivitas termal pada *tube* (W/ m °C)

Koefisien perpindahan panas pada sisi *tube* dapat dihitung dengan persamaan *CollBurn*.

$$h_t = 0,023 Re_t^{0,8} Pr_t^{0,4} \left(\frac{K_t}{d_i} \right) \left(\frac{\mu_t}{\mu_w} \right)^{0,14} \quad (3.43)$$

Dimana,

h_t = koefisien perpindahan panas pada sisi *tube* (W/m² °C)

3.2.3 Perhitungan Koefisien Perpindahan Panas Keseluruhan

Koefisien perpindahan panas keseluruhan dapat dihitung dengan persamaan berikut

$$cond = \frac{d_o \left(\ln \left(\frac{d_o}{d_i} \right) \right)}{2 K_{cs}} \quad (3.44)$$

$$\frac{1}{U_f} = \frac{d_o}{d_i h_t} + \frac{d_o R_{ft}}{d_i} + cond + R_{fs} + \frac{1}{h_s} \quad (3.45)$$

Dimana,

U_f = koefisien perpindahan panas keseluruhan pada kondisi *fouling* (W/m² °C)

$Cond$ = perpindahan panas konduksi pada *tube* (m² °C /W)

K_{cs} = konduktifitas termal material *tube* (W/ m °C)

R_{fs} = *fouling resistance* pada sisi *shell* (m² °C /W)

R_{ft} = *fouling resistance* pada sisi *tube* (m² °C /W)

Persamaan nilai *heat duty*

$$Q = U_f LMTD_{corr.} A_o \quad (3.46)$$

Dimana,

U_f = koefisien perpindahan panas keseluruhan pada kondisi *fouling* (W/m² °C)

$LMTD_{corr.}$ = *Log Mean Temperature Difference corrected* (°C)

A_o = luas permukaan perpindahan panas (m²)

Log Mean Temperature Difference corrected ($LMTD_{corr.}$) dihitung dengan persamaan berikut.

$$LMTD_{corr.} = LMTD \cdot F \quad (3.47)$$

Dimana,

$LMTD$ = Log Mean Temperature Difference (°C)

F = faktor koreksi

Log Mean Temperature Difference (LMTD) merupakan rata-rata perbedaan temperatur antara fluida panas dan fluida dingin. $LMTD$ dihitung dengan persamaan berikut.

$$LMTD = \frac{(T_{h,i} - T_{c,o}) - (T_{h,o} - T_{c,i})}{\ln \left(\frac{T_{h,i} - T_{c,o}}{T_{h,o} - T_{c,i}} \right)} \quad (3.48)$$

Faktor koreksi (F) dihitung dengan persamaan berikut.

$$F = \frac{\sqrt{R^2 + 1} \ln(1 - S) / (1 - RS)}{(R - 1) \ln \frac{2 - S(R + 1 - \sqrt{R^2 + 1})}{2 - S(R + 1 + \sqrt{R^2 + 1})}} \quad (3.49)$$

Dengan nilai R didapatkan dari persamaan berikut

$$R = \frac{T_{h,i} - T_{h,o}}{T_{c,o} - T_{c,i}} \quad (3.50)$$

dan nilai S didapatkan dari persamaan berikut

$$S = \frac{T_{c,o} - T_{c,i}}{T_{h,i} - T_{c,i}} \quad (3.51)$$

Dimana,

R, S = faktor efisiensi temperatur

$T_{c,i}$ = temperatur masuk fluida dingin

$T_{c,o}$ = temperatur keluar fluida dingin

$T_{h,i}$ = temperatur masuk fluida panas

$T_{h,o}$ = temperatur keluar fluida panas

Luas permukaan perpindahan panas dihitung dengan persamaan berikut.

$$A_o = \frac{22}{7} d_o L_{ti} N_t N_s \quad (3.52)$$

3.2.4 Perhitungan *Pressure Drop*

Pressure drop pada sisi *tube* dihitung dengan *Pethukov and Popov's Methods* [15].

$$\Delta P_t = \left[2 \times f \frac{L \times N_p}{d_i} + 2 \times N_p \right] \times \rho_t \times V^2 \quad (3.53)$$

$$f = (1,58 \ln Re_t - 3,28)^{-2} \quad (3.54)$$

Dimana,

ΔP_t = *pressure drop* sisi *tube* (kg/cm²)

f = faktor friksi

L = panjang *tube* (m)

ρ_t = kalor jenis pada *tube* (kg/m³)

V = kecepatan fluida (m/s)

Pressure drop pada sisi *shell* dihitung dengan menggunakan *Bell Delaware's Methods* [3].

$$\Delta P_s = \frac{2 f G_s^2 D_s (N_b + 1)}{\rho_s d_o \left(\frac{\mu_s}{\mu_w} \right)^{0,14}} \quad (3.55)$$

$$f = e^{0,576 - 0,19 \ln Re_s} \quad (3.56)$$

Dimana,

ΔP_s = *pressure drop* sisi *shell* (kg/cm²)

f = faktor friksi

ρ_s = kalor jenis pada *shell* (kg/m³)

3.3 Pemodelan *Fouling Resistance*

Untuk menghitung *fouling resistance* pada *shell and tube heat exchanger* menggunakan Polley model sebagai berikut.

$$\frac{dR_f}{dt} = \alpha Re^{-0,8} Pr^{-\frac{1}{3}} \exp\left(-\frac{E_a}{RT_w}\right) - \gamma Re^{0,8} \quad (3.57)$$

Dimana,

$$\begin{aligned} \frac{dR_f}{dt} &= \text{fouling resistance (m}^2 \text{ K/J)} \\ T_w &= \text{temperatur tube wall (K)} \\ \alpha &= 277,8 \text{ m}^2 \text{ K/J} \\ E_a &= 48 \text{ KJ/mol} \\ \gamma &= 4,17 \cdot 10^{-13} \text{ m}^2 \text{ K/J} \\ R &= 0,008314 \text{ KJ/mol K} \end{aligned}$$

Temperatur *tube wall* dihitung dengan persamaan berikut.

$$T_w = T_{t,av} + \frac{T_{s,av} - T_{t,av}}{1 + \frac{h_t}{h_s}} \quad (3.58)$$

Dimana,

$$\begin{aligned} T_{t,av} &= \text{temperatur rata-rata pada tube (}^{\circ}\text{C)} \\ T_{s,av} &= \text{temperatur rata-rata pada shell (}^{\circ}\text{C)} \end{aligned}$$

3.4 Validasi Pemodelan *Heat Exchanger* dan *Fouling Resistance*

Validasi digunakan untuk menentukan seberapa valid hasil pemodelan yang telah dihitung. Proses validasi dilakukan dengan cara membandingkan hasil pemodelan dengan data *sheet* atau data desain *heat exchanger*. Data desain tersebut dianggap data yang paling benar yang sesuai dengan kondisi yang ada di lapangan, sehingga data desain dijadikan pedoman untuk melakukan validasi. Hasil pemodelan dikatakan valid apabila variasi antara hasil pemodelan dengan data *sheet* kecil atau sama. Apabila hasil pemodelan telah valid kemudian lanjut ke proses berikutnya. Apabila hasil pemodelan belum valid, maka perlu untuk memodelkan ulang *heat exchanger* dan *fouling resistance*.

3.5 Formulasi Fungsi Tujuan

Pada tahap ini, dilakukan penentuan fungsi tujuan optimisasi. Fungsi tujuan merepresentasikan tujuan dilakukannya optimisasi. Fungsi tujuan pada optimisasi ini adalah *fouling resistance* yang minimal pada *shell and tube heat exchanger*.

3.6 Optimisasi *Fouling Resistance* pada *Shell and Tube Heat Exchanger*

Optimisasi pada penelitian ini dirancang agar mendapatkan desain *heat exchanger* yang optimal. Fungsi tujuan optimisasi adalah *fouling resistance* yang minimal. Variabel yang dioptimisasi adalah diameter dalam *shell* (D_s), diameter luar *tube* (d_o), dan jumlah *baffle* (N_b), dengan konstrain adalah *heat duty* dan *pressure drop*. Optimisasi dilakukan dengan menggunakan metode stokastik yakni *Rain Water Optimization*.

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Pemodelan *Heat Exchanger* dan *Fouling Resistance*

Proses desain *heat exchanger* membutuhkan pemodelan *heat exchanger* dan *fouling resistance*. Pemodelan digunakan untuk mendapatkan nilai koefisien perpindahan panas baik di sisi *shell* (h_s) maupun *tube* (h_t), koefisien perpindahan panas keseluruhan dalam kondisi *fouling* (U_f), *heat duty* (Q), *pressure drop* di sisi *shell* (Δp_s) dan di sisi *tube* (Δp_t), *fouling resistance* di sisi *shell* (R_{fs}) dan di sisi *tube* (R_{ft}). Perhitungan pemodelan *heat exchanger* dan *fouling resistance* menggunakan persamaan pada bab 2 dan bab 3.

Perhitungan tersebut membutuhkan spesifikasi *heat exchanger*. Spesifikasi merupakan input yang nantinya akan digunakan untuk pemodelan dan proses optimisasi. Spesifikasi yang dibutuhkan diantaranya adalah dimensi *heat exchanger*, kondisi operasi dan *properties* fluida.

Properties fluida mempengaruhi proses pembentukan *fouling* pada *heat exchanger*. Proses desain *heat exchanger* memerlukan data *properties* fluida sehingga dapat ditentukan geometri yang optimal sesuai dengan fluida yang mengalir pada sisi *shell* dan *tube*. Fluida yang memiliki kecepatan tinggi dapat memperlambat proses terbentuknya *fouling*.

Tabel 4.1 *Properties fluida pada shell and tube heat exchanger*

	<i>Shell</i>		<i>Tube</i>	
	Inlet	Outlet	Inlet	Outlet
Fluida	<i>MP Steam</i>		<i>Main Column Bottom</i>	
Aliran Masa (kg/s)	10,773	10,773	35,368	35,368
Temperatur (°C)	214	320	359	325
Massa jenis (kg/m ³)	10,294	7,729	860,8	884,5
Viskositas (kg/ m s)	0,0000162	0,0000209	0,0005	0,0006
Kalor jenis (kcal/kg °C)	0,766	0,544	0,58	0,56
Konduktivitas termal (kcal/s m °C)	0,00001003	0,00001144	0,00001917	0,00002033
kecepatan (m/s)			1,39	
<i>pressure drop</i> (kg/cm ²)	0,352		0,352	

Apabila semua data telah lengkap, kemudian dilakukan pemodelan *heat exchanger* dan *fouling resistance*. Persamaan pemodelan *heat exchanger* pada bab 2 dan bab 3 disesuaikan dengan jenis dan spesifikasi *heat exchanger* yang telah ditentukan sebelumnya. Hasil pemodelan yang telah dihitung kemudian dibandingkan dengan data desain atau data *sheet heat exchanger*. Berikut adalah hasil pemodelan *heat exchanger* dan *fouling resistance* dibandingkan dengan data desain *heat exchanger*.

Tabel 4.2 Hasil pemodelan *heat exchanger* dan *fouling resistance*

Variabel	Satuan	Desain	Kalkulasi	Variasi
h_s	W/m ² °C	796,25	815,08	2%
h_t	W/m ² °C	1536,46	1536,02	0%
U_f	W/m ² °C	338,11	338,21	0%
A_o	m ²	102,07	101,67	0%
Q	MW	2,05	2,05	0%
Δp_s	psi	4,98	4,59	-8%
Δp_t	psi	4,98	4,58	-8%
R_{ft}	m ² °C/W	0,00052	0,00052	0%
R_{fs}	m ² °C/W	0,00009	0,00009	0%

Tabel di atas merupakan hasil pemodelan *heat exchanger* dan *fouling resistance* yang telah dibandingkan dengan data desain *heat exchanger*. Variasi merupakan perbedaan antara hasil pemodelan dengan data desain *heat exchanger*, sehingga dapat diketahui seberapa valid hasil pemodelan yang telah dihitung. Koefisien perpindahan panas pada sisi *shell* (h_s) memiliki variasi sebesar 2% hal ini dipengaruhi oleh faktor koreksi di sisi *shell*. Sedangkan nilai *pressure drop* sisi *shell* (Δp_s) dan *tube* (Δp_t) memiliki variasi 8% lebih rendah dari data desain karena dipengaruhi oleh faktor friksi yakni gesekan antara fluida dengan pipa baik pada sisi *shell* maupun *tube*.

4.2 Perancangan dan Pengujian Optimisasi

Optimisasi adalah proses untuk memperoleh nilai yang optimal sesuai dengan fungsi tujuan. Proses optimisasi dilakukan dengan melalui beberapa tahap, diantaranya adalah menentukan fungsi tujuan optimisasi, menentukan variabel optimisasi, menentukan *properties* optimisasi dan melakukan optimisasi sistem. Berikut adalah penjelasan lebih lanjut proses optimisasi.

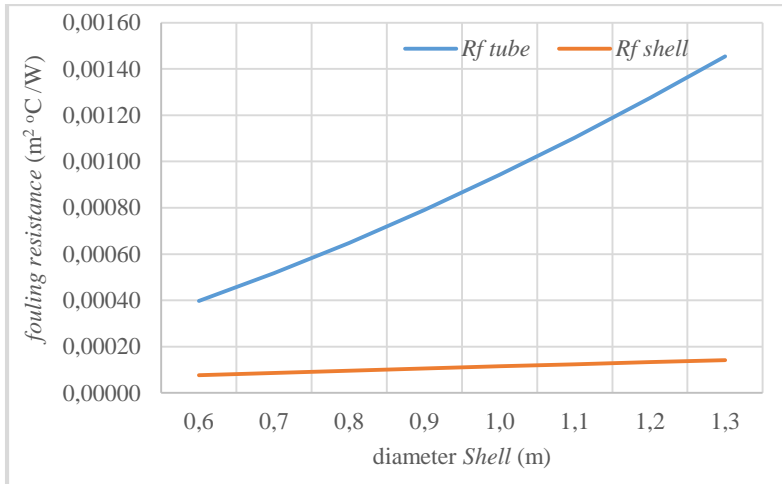
4.2.1 Penentuan Fungsi Tujuan

Sesuai dengan permasalahan yang diangkat dalam tugas akhir ini, maka fungsi tujuan optimisasi adalah *fouling resistance* yang minimal baik pada sisi *shell* maupun *tube*.

4.2.2 Penentuan Variabel yang Dioptimisasi

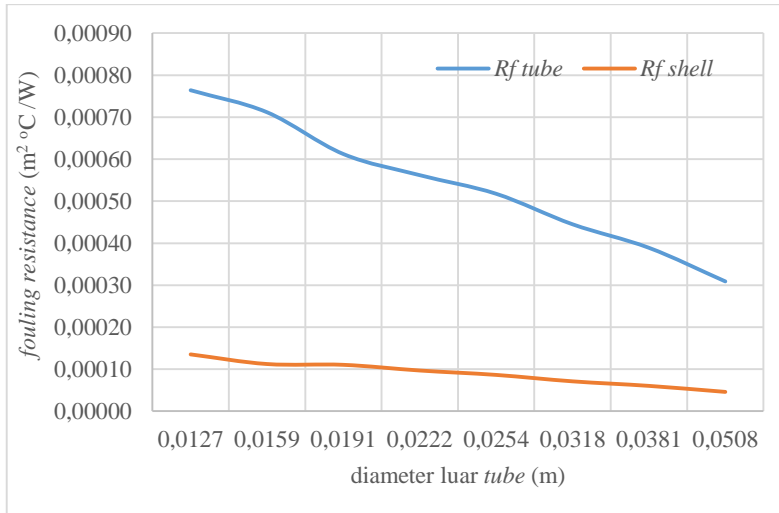
Variabel yang mempengaruhi *fouling resistance* perlu dipertimbangkan untuk mendapatkan *fouling resistance* yang minimal. Variabel-variabel tersebut diperoleh melalui proses pemodelan *heat exchanger* dan *fouling resistance*. Variabel-variabel yang mempengaruhi desain *shell and tube heat exchanger* adalah diameter dalam *shell* (D_s), diameter luar *tube* (d_o), dan jumlah *baffle* (N_b). Variabel-variabel tersebut dijadikan sebagai variabel yang dioptimisasi.

Pengaruh variabel yang dioptimisasi terhadap *fouling resistance* dapat dilihat pada grafik di bawah ini.



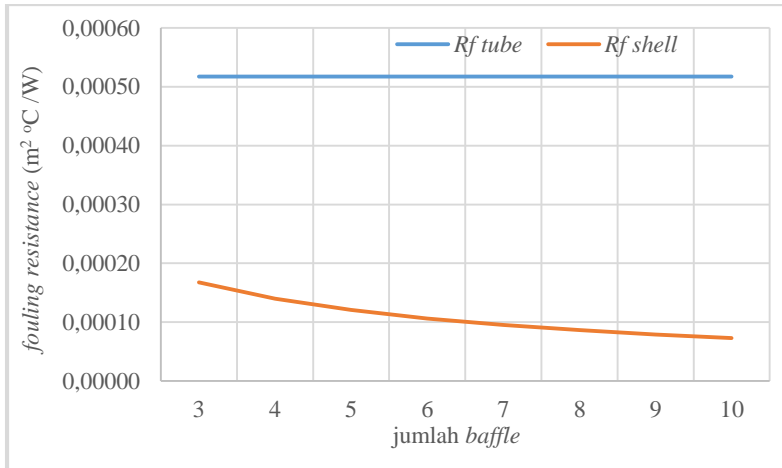
Gambar 4.1 Pengaruh diameter dalam *shell* terhadap *fouling resistance*

Diameter dalam *shell* mempengaruhi *fouling resistance* di *shell* dan di *tube*. Dari grafik diatas dapat dilihat bahwa *fouling resistance* akan semakin rendah dengan nilai diameter dalam *shell* yang semakin kecil. Diameter dalam *shell* menyebabkan perubahan pada luas permukaan baik di sisi *shell* maupun di sisi *tube*. Semakin kecil diameter dalam *shell* maka luas permukaan pada sisi *shell* dan *tube* akan semakin kecil juga, sehingga kecepatan aliran massa pada *shell* dan *tube* semakin tinggi. Hal ini akan menyebabkan, bilangan *Reynold* tinggi dan menyebabkan *fouling resistance* semakin rendah.



Gambar 4.2 Pengaruh diameter luar *tube* terhadap *fouling resistance*

Diameter luar *tube* mempengaruhi *fouling resistance* di sisi *shell* dan *tube*. Dari grafik di atas dapat dilihat bahwa semakin besar diameter luar *tube* maka *fouling resistance* akan semakin rendah. Hal ini dikarenakan diameter dalam *tube* memiliki pengaruh langsung pada bilangan *Reynold* baik di sisi *shell* maupun *tube*. Semakin besar diameter dalam *tube* menyebabkan bilangan *Reynold* juga akan semakin besar. Dan membentuk aliran turbulen dengan kecepatan fluida yang tinggi. Persamaan *fouling* dengan menggunakan *Polley model* bergantung pada perubahan bilangan *Reynold*, semakin besar bilangan *Reynold* maka *fouling resistance* akan semakin rendah. Hal ini berarti proses penggerusan *fouling* lebih cepat daripada proses pembentukannya.



Gambar 4.3 Pengaruh jumlah *baffle* terhadap *fouling resistance*

Jumlah *baffle* hanya mempengaruhi *fouling resistance* pada sisi *shell* dan tidak mempengaruhi sisi *tube*. Dari grafik di atas dapat dilihat bahwa semakin banyak jumlah *baffle* maka *fouling resistance* akan semakin rendah. Hal ini sesuai dengan fungsi *baffle* yakni sekat untuk mengatur aliran dalam *shell* agar terbentuk aliran turbulen. Aliran turbulen disebabkan karena jarak antar *baffle* yang semakin kecil sehingga luas permukaan pada *shell* menjadi lebih kecil. Ketika luas permukaan pada *shell* menjadi lebih kecil maka kecepatan aliran massa semakin tinggi dan bilangan *Reynold* juga akan tinggi sehingga *fouling resistance* semakin rendah.

4.2.3 Penentuan *Properties* pada Teknik Optimisasi

Pemilihan algoritma menjadi suatu hal yang harus dipertimbangkan dalam melakukan optimisasi. Pemilihan algoritma dapat ditentukan berdasarkan fungsi tujuan dan pemodelannya. Karena pemodelan *heat exchanger* pada optimisasi ini adalah non linier yang memiliki beberapa lokal optimum, maka teknik optimisasi membutuhkan algoritma yang dapat menemukan global optimum. Optimisasi dilakukan dengan menggunakan

metode stokastik yakni *Rain Water Optimization*. Properti yang dibutuhkan dalam optimisasi diantaranya sebagai berikut:

a. Jumlah populasi

Jumlah populasi menentukan jumlah *rain water* yang jatuh ke bumi yang akan terlibat dalam proses optimisasi. Jumlah populasi yang digunakan adalah 200.

b. Jumlah variabel yang dioptimisasi

Variabel yang mempengaruhi fungsi tujuan yakni diameter dalam *shell* (D_s), diameter luar *tube* (d_o), dan jumlah *baffle* (N_b), maka jumlah variabel yang dioptimisasi adalah 3.

c. Batas atas dan batas bawah optimisasi

Batas atas dan batas bawah optimisasi digunakan sebagai jangkauan untuk mengacak nilai-nilai variabel optimisasi agar memenuhi fungsi tujuan. Batas atas dan batas bawah diperoleh berdasarkan standar TEMA yang disesuaikan dengan pemodelan *heat exchanger*. Batas atas dan batas bawah variabel optimisasi sebagai berikut :

- diameter dalam *shell* = 0,5 m – 0,9 m
- diameter luar *tube* = 0,0127 m – 0,02224 m
- jumlah *baffle* = 4 – 7

d. Iterasi

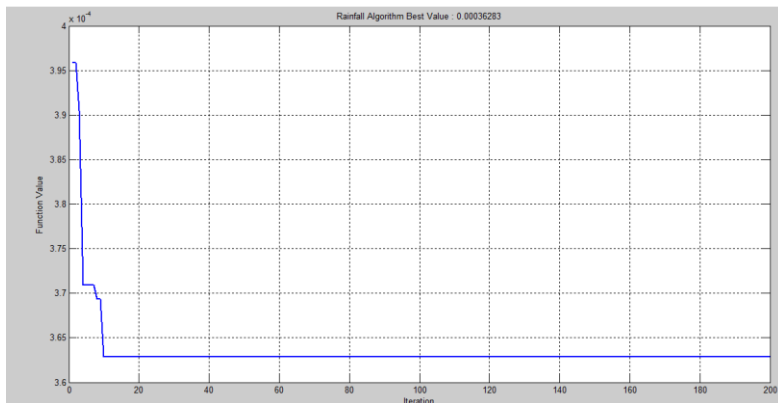
Iterasi digunakan untuk menentukan jumlah perputaran proses optimisasi. Pada tugas akhir ini digunakan 200 kali iterasi. Hal ini ditentukan berdasarkan grafik hasil optimisasi pada *software* MATLAB yang telah menunjukkan nilai yang konstan pada iterasi ke 200. Apabila grafik telah konstan dan hasil optimisasi tidak berubah, maka iterasi telah selesai.

4.2.4 Optimisasi *Fouling Resistance* pada *Shell and Tube Heat Exchanger*

Optimisasi *fouling resistance* perlu menentukan konstrain yakni batasan yang harus dipenuhi agar kinerja *heat exchanger*

efisien. Konstran pada optimisasi ini yakni nilai *heat duty* (Q) pada *shell and tube heat exchanger* harus sesuai dengan data desain *heat exchanger* yakni sebesar 2,05 MW. *Heat duty* adalah panas yang harus diserap oleh *heat exchanger*. Apabila nilai *heat duty* tidak terpenuhi maka efisiensi *heat exchanger* menjadi tidak maksimal. Konstran yang lainnya yakni *pressure drop* pada *shell* (Δp_s) dan *tube* (Δp_t) yang nilainya juga tidak boleh melebihi data desain yakni sebesar 4,5 psi. Apabila nilai *pressure drop* melebihi nilai data desain *heat exchanger* maka berpengaruh pada konsumsi pompa yang membutuhkan energi lebih besar.

Semakin kecil diameter luar *tube* maka *pressure drop* pada sisi *shell* dan *tube* akan semakin besar, semakin kecil nilai diameter dalam *shell* maka nilai *pressure drop* semakin besar, semakin banyak jumlah *baffle* maka *pressure drop* di sisi *shell* akan semakin besar namun jumlah *baffle* tidak berpengaruh pada *pressure drop* di sisi *tube*. Fungsi tujuan, variabel optimisasi dan konstran merupakan komponen utama dalam proses optimisasi. Berikut adalah grafik hasil optimisasi *fouling resistance* menggunakan *Rain Water Optimization*.



Gambar 4.4 Grafik hasil optimisasi *fouling resistance* pada software MATLAB

Optimisasi sistem dilakukan menggunakan *software* MATLAB yakni dengan cara menginputkan pemodelan *heat exchanger* dan *fouling resistance*, fungsi tujuan optimisasi, *properties* optimisasi dan algoritma *rain water*. Grafik tersebut menunjukkan hasil *fouling resistance* yang paling minimal. Berikut adalah tabel perbandingan geometri *heat exchanger* dan *fouling resistance* sebelum dan setelah dioptimisasi.

Tabel 4.3 Perbandingan geometri *heat exchanger* dan *fouling resistance* sebelum dan setelah dioptimisasi

Variabel	Satuan	Desain	Optimisasi	Perbedaan
d_o	m	0,0254	0,02223	-12%
n_b		8	7	-13%
D_s	m	0,7	0,6	-14%
d_i	m	0,02	0,02	-16%
N_t		131	125	-5%
l_{tp}	m	0,03	0,03	-12%
L_{bc}	m	0,50	0,56	13%
S_m	m ²	0,07	0,07	-3%
G_s	kg/ s m ²	74,54	66,76	-10%
R_{es}		102066	80005	-22%
h_s	W/m ² °C	815,08	845,86	4%
A_t	m ²	0,04	0,03	-33%
G_t	kg/ s m ²	1202,23	1788,30	49%
R_{et}		46272	57844	25%
h_t	W/m ² °C	1536,02	2185,10	42%
U_f	W/m ² °C	338,21	416,36	23%
A_o	m ²	101,67	82,52	-19%

Variabel	Satuan	Desain	Optimisasi	Perbedaan
Q	MW	2,05	2,05	0%
Δp_s	psi	4,98	4,03	-19%
Δp_t	psi	4,98	4,31	-13%
R_{ft}	$\frac{m^2}{^\circ C/W}$	0,00052	0,00029	-44%
R_{fs}	$\frac{m^2}{^\circ C/W}$	0,00009	0,00007	-18%

Tabel di atas menunjukkan bahwa untuk mencapai *fouling resistance* yang minimal maka diperoleh nilai diameter dalam *shell* (D_s) yang lebih kecil dari data desain yakni 0,6 m, diameter luar *tube* (d_o) yang lebih kecil dari data desain yakni sebesar 0,02223 m dan jumlah *baffle* (N_b) yang lebih sedikit yakni 7 buah. Desain yang telah dioptimisasi tersebut memiliki *fouling resistance* pada *shell* (R_{fs}) lebih rendah 18 % dari data desain, sedangkan pada sisi *tube* (R_{ft}) lebih rendah 44 % dari data desain. Nilai koefisien perpindahan panas keseluruhan pada kondisi *fouling* (U_f) meningkat sebesar 23% dengan luas *heat exchanger* (A_o) lebih kecil 19%. Dari hasil di atas maka, perubahan geometri *heat exchanger* dapat mengurangi *fouling resistance* dan meningkatkan nilai koefisien perpindahan panas keseluruhan, geometri *heat exchanger* menjadi lebih kecil, dengan nilai *heat duty* yang tetap dan nilai *pressure drop* yang lebih rendah data desain *heat exchanger*. Hal ini berarti *fouling resistance* dapat diminimalkan dengan mengoptimalkan geometri *heat exchanger*.

4.2.5 Optimisasi *Fouling Resistance* pada *Shell and Tube Heat Exchanger* Menggunakan Beberapa Metode Stokastik

Optimisasi *fouling resistance* pada penelitian ini menggunakan *Rain Water Algorithm* (RWA). Untuk membandingkan hasil optimisasi *fouling resistance* maka digunakan beberapa metode stokastik yang lain yakni *Killer Whale Algorithm* (KWA), *Duelist Algorithm* (DA), *Particle Swarm Optimization* (PSO), dan *Genetic Algorithm* (GA). Dari beberapa

metode optimisasi yang digunakan, maka dapat dilihat pada tabel 4.4 bahwa hasil optimisasi menggunakan *RWA*, *KWA* dan *PSO* memiliki hasil optimisasi yang hampir sama. Sedangkan untuk *GA* dan *DA* menghasilkan hasil optimisasi yang berbeda. Hal ini dipengaruhi oleh fungsi tujuan optimisasi. Apabila fungsi tujuan optimisasi adalah minimum maka metode optimisasi *RWA*, *KWA* dan *PSO* memberikan hasil yang lebih baik daripada *GA* dan *DA*. Berikut adalah perbandingan hasil optimisasi *fouling resistance* pada *shell and tube heat exchanger* menggunakan beberapa metode stokastik.

Tabel 4.4 Perbandingan hasil optimisasi *fouling resistance* menggunakan beberapa metode stokastik

variabel	satuan	desain	hasil optimisasi				
			<i>RWA</i>	<i>KWA</i>	<i>PSO</i>	<i>GA</i>	<i>DA</i>
d_o	m	0,0254	0,0222	0,0222	0,0222	0,0219	0,0194
n_b		8,00	6,95	7	7	6,76	6,68
D_s	m	0,70	0,6	0,6	0,6	0,6	0,52
U_f	W/m ² °C	338,21	416,26	416,36	416,36	415,98	483,85
A_o	m ²	101,67	82,48	82,52	82,48	83,25	69,67
Q	MW	2,05	2,05	2,05	2,05	2,07	2,01
Δp_s	psi	4,98	4,03	4,03	4,03	3,76	4,64
Δp_t	psi	4,98	4,31	4,31	4,31	4,37	4,69
R_{ft}	m ² °C / W	0,00052	0,00029	0,00029	0,00029	0,00029	0,00025
R_{fs}	m ² °C / W	0,00009	0,00007	0,00007	0,00007	0,00007	0,00007

Halaman ini sengaja dikosongkan

BAB V PENUTUP

5.1 Kesimpulan

Optimisasi *fouling resistance* pada desain *shell and tube heat exchanger* menggunakan teknik optimisasi stokastik dengan hasil sebagai berikut.

- a. Geometri yang optimal agar diperoleh *fouling resistance* yang minimal adalah diameter dalam *shell* (D_s) sebesar 0,6 m, diameter luar *tube* (d_o) sebesar 0,02223 m dan jumlah *baffle* (n_b) sebanyak 7 buah.
- b. *Fouling resistance* pada sisi *shell* (R_{fs}) dan *tube* (R_{ft}) berturut-turut berkurang sebesar 18 % dan 44 % dibandingkan sebelum dioptimisasi. Nilai R_{fs} dan R_{ft} setelah dioptimisasi sebesar $0,00029 \text{ m}^2\text{°C/W}$ dan $0,00007 \text{ m}^2\text{°C/W}$.
- c. Penurunan *fouling resistance* menyebabkan nilai koefisien perpindahan panas keseluruhan dalam keadaan *fouling* (U_f) meningkat sebesar 23 % dengan nilai $416,36 \text{ W/m}^2\text{°C}$.
- d. Penurunan *fouling resistance* menyebabkan luas permukaan *heat exchanger* menjadi lebih kecil yakni berkurang sebesar 19 % dengan nilai $82,52 \text{ m}^2$.
- e. Penurunan *fouling resistance* menyebabkan *pressure drop* turun yakni pada *tube* berkurang sebesar 13 % dengan nilai 4,03 psi dan pada *shell* berkurang sebesar 19 % dengan nilai 4,31 psi
- f. *Fouling resistance* disebabkan karena pemilihan geometri *heat exchanger*. Sehingga *fouling resistance* dan pemilihan geometri *heat exchanger* merupakan hubungan sebab akibat yakni proses desain *heat exchanger* berpengaruh pada *fouling resistance*
- g. Teknik optimisasi menggunakan metode stokastik yakni *RWA* memiliki hasil optimisasi yang optimal jika dibandingkan dengan hasil optimisasi menggunakan metode stokastik yang lain seperti *KWA*, *PSO*, *GA* dan *DA*.

5.2 Saran

Dari hasil tugas akhir ini dapat diberikan beberapa saran untuk pengembangan penelitian selanjutnya antara lain :

- a. *Heat exchanger* yang digunakan berjumlah lebih dari satu dengan spesifikasi *heat exchanger* yang berbeda.
- b. *Fouling resistance* dimodelkan dengan menggunakan persamaan yang lain, selain *Polley model*.
- c. *Properties* fluida yang digunakan beragam.
- d. *Fouling resistance* melibatkan dua faktor yakni perubahan pada kondisi operasi dan geometri *heat exchanger*.


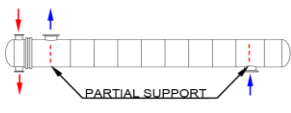
DAFTAR PUSTAKA

- [1] R. M. Thirumarimurugan, "*Performance Analysis of Shell and Tube Heat Exchanger Using Miscible System*," *American Journal of Applied Sciences*, vol. 5, no. 5, pp. 548-552, 2008.
- [2] H. L. S. Kakac, *Heat Exchangers Selection, Rating, and Thermal Design*, London: CRC Press, 2001.
- [3] K. Thulukkanam, *Heat Exchanger Design Handbook*, New York: CRC Press, 2013.
- [4] Ratel M., "*Investigation of Fouling Rates in A Heat Exchanger Using An Innovative Fouling Rig*," pp. 36-41, 2013.
- [5] Wilson D.I, Polley G.T., Pugh S.J, "*Ten Years of Ebert, Panchal and The "Threshold Fouling" Concept*," vol. 6, pp. 25-35, 2005.
- [6] Kern D.Q, *Process Heat Transfer*, New York: Mc Graw-Hill Book Company, 1965.
- [7] R. Subramanian, "*Shell-and-Tube Heat Exchangers*," pp. 1-10, 2010.
- [8] Richard C.B., *Standards of The Tubular Exchanger Manufactures Association*, New York: Tubular Exchanger Manufacturers Association, 2007.
- [9] D. Massada, "*Evaluation of Lumped Parameter Heat*," *ASME*, vol. 1, no. 1, pp. 1-10, 1982.
- [10] Tangient LLC, "*Heat Exchanger Crude Oil Fouling and Fouling Control*," San Francisco, 2015.
- [11] PT.PERTAMINA RU-IV, "*Heat Exchanger 101-E-527A*," PT.PERTAMINA RU IV, Cilacap, Jawa Tengah, Indonesia, 2017.
- [12] Nasr M.R.J., "*Application of Threshold Model with Various Tube Wall Temperatures for Crude Oil Preheat Train Fouling*," vol. 25, no. 3, pp. 49-58, 2006.

- [13] Costa A.L.H, "*Parameter Estimation of Fouling Models in Crude Preheat Trains*," pp. 39-46, 2011.
- [14] T. R. Biyanto, "*Rain Water Optimization Algorithm*," Process Design, Control and Optimization Laboratory, Department Engineering Physics, Institut Teknologi Sepuluh Nopember, Surabaya, 2017.

LAMPIRAN

A. Data Sheet Heat Exchanger

 GS Engineering & Construction		PJT.NO. 101830 ITEM NO. 101-E-527A/B		SHELL & TUBE HEAT EXCHANGER DATA SHEET (MKH UNIT)				SHEET 1 OF 2	
PROJECT CILACAP RFCC PROJECT CLIENT PT. PERTAMIT PROCESSING LOCATION CILACAP, CENTRAL JAVA, INDONESIA SERVICE MAIN COLUMN BOTTOMS-MP STEAM SUPERHEATERS MFR. NAME APPLIED CODE ASME SEC. VIII, DIV.1 TEMA CLASS R		DOC.NO. : P & I.D NO. : REV. DATE PREP'D CHK'D BY APP'D 0 2012-01-06 K.H.JEONG H.J. LEE 1 2012-06-05 K.H.JEONG H.J. LEE 2 2013-02-14 K.H.JEONG H.J. LEE							
TYPE BEU SIZE 700 mm X 5000 mm SURFACE / UNIT (EFF.) 204.13 m ²		INSTALL TYPE CONNECTED IN 2 PARALLEL 1 SERIES SHELL / UNIT TWO(2) SURFACE / SHELL (EFF.) 102.07 m ²							
PERFORMANCE OF ONE UNIT (OPERATING CASE NAME : DESIGN CASE)									
FLUID ALLOCATION				SHELL SIDE		TUBE SIDE			
IN OUT				IN OUT		IN OUT			
FLUID CIRCULATED				MP STEAM		MAIN COLUMN BOTTOMS			
FLUID QUANTITY - TOTAL kg/h				38782 x 1.38		127323 x 1.38			
- kg/h				38782 x 1.38		127323 x 1.38			
- kg/h				38782 x 1.38		127323 x 1.38			
- kg/h				38782 x 1.38		127323 x 1.38			
- NONCONDENSABLE kg/h				38782 x 1.38		127323 x 1.38			
TEMPERATURE deg-C				214 320		359 325			
DENSITY VAP. / LIQ. kg/m ³				10.294 7.7298		860.8 884.5			
VISCOSITY VAP. / LIQ. cP				0.0162 0.0209		0.457 0.575			
MOLE. WEIGHT				VAPOR		NONCONDENSABLE			
SPECIFIC HEAT VAP. / LIQ. kcal/kg-deg-C				0.7659 0.5441		0.58 0.58			
THERMAL CONDUCTIVITY VAP. / LIQ. Kcal/hr-m-deg-C				0.0361 0.0412		0.069 0.0732			
LATENT HEAT kcal/kg				19.90		6.124			
INLET PRESSURE kg/cm ² g				19.90		1.39			
VELOCITY m/s				0.352 / 0.35		0.352 / 0.35			
PRESSURE DROP (allw./cal.) kg/cm ²				0.352 / 0.35		0.352 / 0.35			
FOULING RESIST.(min.) m ² -hr-deg-C/Kcal				0.0001		0.0006			
FILM HEAT TRANSFER COEFF. Kcal/m ² -hr-deg-C				684.65		1321.12			
HEAT EXCHANGED 2.49 x 1.38 mm kcal/hr				MTD, (CORRECTED) 60.0 deg_C		290.72 CLEAN 388.75			
TRANSFER RATE Kcal/m ² -hr-deg-C				SERVICE 280.47 CALCULATED 290.72		CLEAN 388.75			
CONSTRUCTION OF ONE SHELL									
SHELL SIDE				TUBE SIDE		SKETCH (BUNDLE / NOZ. ORIENT)			
DESIGN PRESS. kg/cm ² g 24.5 / F.V. @ 120 deg-c				19					
TEST PRESS. kg/cm ² g 370 / 18 (MDMT)				400 / 18 (MDMT)		PARTIAL SUPPORT			
DESIGN TEMP. deg-C 1				FOUR(4)		INSULAT'N B/M, m ² : S/S= 13.5, T/S= 2.2			
NO. OF PASS PER SHELL 3.0				(NOTE. 1)		PAINT'G B/M, m ² : S/S= 14.7, T/S= 3.1, SUP= 1.5			
CORR. ALLOW. mm				/		/			
CORR./ERO. CAUSED BY				/		/			
INSUL. (TYPE / THK, mm)				/		/			
NOZZLE INLET 10 inch / 1 / 300# - RF				6 inch / 1 / 300# - RF		/			
(SIZE / NO. OUTLET 10 inch / 1 / 300# - RF				6 inch / 1 / 300# - RF		/			
/ REMARK				/		/			
INTERMEDIATE				/		/			
MATERIAL									
TUBE TYPE 410				SHELL		Killed Carbon Steel			
CHANNEL OR BONNET Killed Carbon Steel (NOTE. 1)				SHELL COVER		Killed Carbon Steel			
TUBESHEET-STATIONARY Killed Carbon Steel (NOTE. 1)				CHANNEL COVER		Killed Carbon Steel (NOTE. 1)			
FLOATING HEAD COVER				TUBESHEET-FLOATING		/			
GASKET (SHELL / CHANNEL / FLOATING H)				/		/			
TUBE									
- NO. 131-U - O.D 25.40 mm				- THK (Avg.) 2.77 mm		- LENGTH 5000 mm		- PITCH 31.75mm & 90 deg	
- TUBE TYPE : Plain				- TUBE TO TUBESHEET JOINT		STRENGTH WELD WITH LIGHT EXPANSION			
- I.D 700 mm				- SHELL COVER : INTEGRAL		IMPINGEMENT PROTECTION YES		YES	
BAFFLES (CROSS) SINGLE-SEG.				- NO. 8		- %CUT (DIA.) 26.4		- VERTICAL	
- SPACING (mm)				- INLET 600		- OUTLET		/	
- LENGTH mm				- SEAL TYPE :		U-BEND SUPPORT TYPE FULL SUPPORT PLATE		/	
BYPASS SEAL ARRANGE. - SEAL STRIPS 2 PAIRS				DESIGN M.M.T. (deg-C)		- SEAL RODS - EA		/	
EXPANSION JOINT (Y/N) - TYPE:				Bundle Entrance 2291.42		- SHELL SIDE - TUBE SIDE		/	
Rho-v ² (kg/m-sec ²) Inlet Nozzle 2311.43				Bundle Exit 967.03		- BUNDLE		kg 3,031	
WEIGHT/SHELL - SHIPPING kg 5,461				- FILLED W / WATER kg 7,677		- BUNDLE		kg 3,031	
NOTES									
1. TUBE SIDES LINED WITH A MINIMUM OF 3 mm TYPE 405 OR TYPE 410S.									
2. THE HOT SIDE FLUID CONTAIN 0.76 wt% SULFUR.									
3. PARTIAL SUPPORTS SHALL BE PROVIDED UNDER INLET AND OUTLET NOZZLES.									
4. THE SHELL&TUBE SIDE ARE SUBJECT TO STEAM OUT (F.V@150deg-C)									

APPLIED PROCESS DATASHEET REV. : 2 DATE : 2012-06-04

Global Best & First

Halaman ini sengaja dikosongkan

B. Koding Optimisasi *Fouling Resistance Shell And Tube Heat Exchanger* Menggunakan Metode Stokastik pada *Software MATLAB*

- *Rain Water Algorithm*

```
%% Inisialisasi
% v = velocity
% g = Gravity
% m = mass
% h = ketinggian
% vo = velocity awal
% Ep = 1/2 mv^2
% Ek = mgh
% E = jarak
% alfa = constant for movement update
% a = g alp t
% t = time // constant
% dim = dimension
% N = number of raindrop
% upbound = upper bound
% lowbound = lower bound
% iter = max iteration
% minmax = min / max
clear all; close all; clc;
dim = 3;
N = 200;    %jumlah air
alfa=360;
G=10;      %gravitasi
t = 1;     %time constant
% do=0.019050;
% ds=0.5;
% nb=6;

upbound = [0.9 0.02224 7];
lowbound = [0.6 0.0127 5];
iter = 200;

Rpower=1;
min_flag=1;
```

```

minmax = 1;
Rnorm=2;
convergence_curve=zeros(1,iter);

% Initialize population, position:
if size(upbound,2)==1
    X=rand(N,dim).*(upbound-
lowbound)+lowbound;
end
if size(upbound,2)>1
    for i=1:dim
        high=upbound(i);
        low=lowbound(i);
        X(:,i)=rand(N,1).*(high-low)+low;
    end
end
Bestpos=zeros(1,dim);
Meanpos=zeros(1,dim);
FBest=zeros(1,dim);
LBest=zeros(1,dim);
Eo=zeros(N,dim);
V=zeros(N,dim);
M = zeros(N);
P = 0;
%% Main Program
while P<iter
    for iteration = 1:iter
        %% inisialisasi Search Agent dan Objective
        Function
        [N,dim]=size(X);
        for i=1:N
            %%Agent that go out of the search space,
            are reinitialized randomly .
            Tp=X(i,:)>upbound;
            Tm=X(i,:)<lowbound;

            X(i,:)=(X(i,:).*(~(Tp+Tm)))+( (rand(1,dim).*(
upbound-upbound)+lowbound).*(Tp+Tm));
        end
    end
end

```

```

for i=1:N
    %L is the location of agent number 'i'
    L=X(i,:);
    %calculation of objective function for
    agent number 'i'
    fobj=@(X) (model(X));
    fitness(i)=fobj(X(i,:));
end

if minmax==1
    [best best_X]=min(fitness); %minimization.
else
    [best best_X]=max(fitness); %maximization.
end

if iteration==1
    Fbest=best;Lbest=X(best_X,:);
end
if minmax==1
if best<Fbest    %minimization.
    Fbest=best;Lbest=X(best_X,:);
end
else
if best>Fbest    %maximization
    Fbest=best;Lbest=X(best_X,:);
end
end

Bestpos=[Bestpos Fbest];
Meanpos=[Meanpos mean(fitness)];
%% Hujan jatuh = energi potensial =  $E_p = 1/2$ 
mv^2
% velocity calculation
Fmax=max(fitness); Fmin=min(fitness);
Fmean=mean(fitness);
[i N]=size(fitness);

if Fmax==Fmin

```

```

        vo=ones(N,1);
    else

    if minmax==1 %for minimization
        best=Fmin;worst=Fmax;
    else %for maximization
        best=Fmax;worst=Fmin;
    end

    vo=(fitness-worst)./(best-worst);

end
M= rand(N);
vo=(vo./sum(vo))*M.*t;
% velocity calculation berfungsi untuk
% menentukan butiran hujan yang jatuh
% terlebih dahulu berdasarkan fitness dari
% setiap agents.
%%

% [N,dim]=size(X);
final_per=1.5; %In the last iteration, only
1.5 percent of agents

kbest=final_per+(1-iteration/iter)*(100-
final_per);
kbest=round(N*kbest/100);

[Ms ds]=sort(vo, 'descend');

for i=1:N

    for ii=1:kbest
        j=ds(ii);
        if j~=i
            R=norm(X(i,:)-X(j,:),Rnorm);
%Euclidian distanse.
            for k=1:dim

```

```

Eo(i,k)=Eo(i,k)+rand*(vo(j))*(X(j,k)-
X(i,k))/(R^Rpower+eps));

        end
    end
end

%%acceleration
E = Eo*exp(-alfa*iteration/iter);
a=E.*G;

%movement.
% [N,dim]=size(X);
V=rand(N,dim).*V+a;
X=X+V;

P = P + 1;
convergence_curve(P) = Fbest;
jx=plot((1:iter),convergence_curve,'LineWidth
h',2); grid on;
title(['Rainfall Algorithm Best Value : '
num2str(Fbest)]);
xlabel('Iteration');
ylabel('Function Value');
end
end

```

- *Killer Whale Algorithm*

```

clc;
clear;
close all;

%% Problem Definition
Dimension = 3;           % dimensi diganti
sesuai dengan jumlah variabel yang dioptimasi
%Constraint

UB = [0.9 0.02224 7]; % Upper Bounds diganti
sesuai dengan constraint fungsi objektif
LB = [0.6 0.0127 5]; % Lower Bounds diganti
sesuai dengan constraint fungsi objektif

CostFunction=@(x) (model(x));           % Cost
Function

nVar=Dimension;           % Number of
Decision Variables

VarSize=[1 nVar];       % Size of Decision
Variables Matrix

VarMin=LB;               % Lower Bound of Variables
VarMax=UB;               % Upper Bound of Variables

%% PSO Parameters

MaxIt=200;               % Maximum Number of
Iterations

nPop=100;                % Population Size (Swarm
Size)
nTeam = 10;              % Team (Number of Leader)
TeamSize = [];

```

```

for i=1:nTeam-1
    TeamSize(i) = ceil(nPop/nTeam);
end
TeamSize(nTeam) = nPop - sum(TeamSize);

% PSO Parameters
w=1;           % Inertia Weight
wdamp=0.99;    % Inertia Weight Damping
Ratio
c1=1.5;        % Personal Learning
Coefficient
c2=2.0;        % Global Learning Coefficient
c3=1.0;        % Leader Influence
Coefficient

Porder=3;      % order of Polynomial

% If you would like to use Constriction
Coefficients for PSO,
% uncomment the following block and comment
the above set of parameters.

% % Constriction Coefficients
% phi1=2.05;
% phi2=2.05;
% phi=phi1+phi2;
% chi=2/(phi-2+sqrt(phi^2-4*phi));
% w=chi;       % Inertia Weight
% wdamp=1;     % Inertia Weight Damping
Ratio
% c1=chi*phi1; % Personal Learning
Coefficient
% c2=chi*phi2; % Global Learning
Coefficient

% Velocity Limits
VelMax=0.1*(VarMax-VarMin);
VelMin=-VelMax;

```

```

%% Initialization

initial_whales.Position=[];
initial_whales.Cost=[];
initial_whales.Velocity=[];
initial_whales.Best.Position=[];
initial_whales.Best.Cost=[];

whales= repmat(initial_whales,nPop,1);
leader_whales=repmat(initial_whales,nTeam,1);
leader_whales_poly = [];
leader_whales_std = [];
temp_whales = [];
tempeval_whales = [];

GlobalBest.Cost=inf;

for i=1:nPop

    % Initialize Position

    whales(i).Position=unifrnd(VarMin,VarMax,VarSize);
    % Initialize Velocity
    whales(i).Velocity=zeros(VarSize);

    % Evaluation

    whales(i).Cost=CostFunction(whales(i).Position);
    % temp_whales(:, i) = whales(i).Position;

end
% tempeval_whales(1:nPop) =
whales.Cost(1:nPop);
for i=1:nPop
%     whales(i).Cost = tempeval_whales(i);
%     Store data for polyfit

```



```

        for j=1:Dimension
            whalesPosition(i,j) =
whales(i).Position(j);
        end
        whalesPosition(i,Dimension+1) =
whales(i).Cost;

        % Update Personal Best

whales(i).Best.Position=whales(i).Position;
whales(i).Best.Cost=whales(i).Cost;

        % Update Global Best
        if whales(i).Best.Cost<GlobalBest.Cost

            GlobalBest=whales(i).Best;

        end

    end

end

    temp_whales = [];
    tempeval_whales = [];
    for t=1:nTeam
        for j=1:Dimension
            leader_whales_std(t,j) =
std2(whalesPosition(((t-
1)*ceil(nPop/nTeam))+1:((t-
1)*ceil(nPop/nTeam))+TeamSize(t),j));
            buffer = polyfit(whalesPosition(((t-
1)*ceil(nPop/nTeam))+1:((t-
1)*ceil(nPop/nTeam))+TeamSize(t),j),whalesPos
ition(((t-1)*ceil(nPop/nTeam))+1:((t-
1)*ceil(nPop/nTeam))+TeamSize(t),Dimension+1)
,Porder);
            leader_whales_poly(t,:,j) = buffer;

            syms x;

```

```

        fun =
matlabFunction(poly2sym(buffer));
        leader_whales(t).Position(j) =
fminsearch((fun),0);

        leader_whales(t).Position(j) =
min(leader_whales(t).Position(j), UB(j));
        leader_whales(t).Position(j) =
max(leader_whales(t).Position(j), LB(j));
    end
    temp_whales(:, t) =
leader_whales(t).Position;

end
% tempeval_whales =
CostFunction(temp_whales);
for t=1:nTeam
    % leader_whales(t).Cost =
tempeval_whales(t);
    leader_whales(t).Cost =
CostFunction(leader_whales(t).Position);
end

BestCost=zeros(MaxIt,1);

%% PSO Main Loop

for it=1:MaxIt

    for t=1:nTeam

        temp_whales = [];
        tempeval_whales = [];

        for i=((t-1)*ceil(nPop/nTeam))+1:((t-
1)*ceil(nPop/nTeam))+TeamSize(t)
            % make into cluster
            % each cluster will have it's own
leader

```

```

        % each member of cluters should
chate their own leader
        % leader get information from
their member, and draw a polynomial
        % equation to map the scanned
area

        % Update Velocity
        % Leader or GlobalBest
        if GlobalBest.Cost <
leader_whales(t).Cost % min
            ct3 = 0;
            ct2 = c2;
        else
            ct3 = c3;
            ct2 = 0;
        end;

        whales(i).Velocity =
w*whales(i).Velocity ...

+c1*rand(VarSize).*(whales(i).Best.Position-
whales(i).Position) ...

+ct2*rand(VarSize).*(GlobalBest.Position-
whales(i).Position) ...

+ct3*rand(VarSize).*(leader_whales(t).Positio
n);

        % Apply Velocity Limits
        whales(i).Velocity =
max(whales(i).Velocity, VelMin);
        whales(i).Velocity =
min(whales(i).Velocity, VelMax);

        % Update Position
        whales(i).Position =
whales(i).Position + whales(i).Velocity;

```

```

        % Velocity Mirror Effect

IsOutside=(whales(i).Position<VarMin |
whales(i).Position>VarMax);
        whales(i).Velocity(IsOutside)=-
whales(i).Velocity(IsOutside);

        % Apply Position Limits
        whales(i).Position =
max(whales(i).Position,VarMin);
        whales(i).Position =
min(whales(i).Position,VarMax);

        temp_whales(:, i) =
whales(i).Position;
        end

        % tempeval_whales =
CostFunction(temp_whales);

        for i=((t-1)*ceil(nPop/nTeam))+1:((t-
1)*ceil(nPop/nTeam))+TeamSize(t)

            % Evaluation
            whales(i).Cost =
CostFunction(whales(i).Position);
            % whales(i).Cost =
tempeval_whales(i);

            % Store data for polyfit
            for j=1:Dimension
                whalesPosition(i,j) =
whales(i).Position(j);
            end
            whalesPosition(i,Dimension+1) =
whales(i).Cost;

            % Update Personal Best

```

```

        if
whales(i).Cost<whales(i).Best.Cost

whales(i).Best.Position=whales(i).Position;

whales(i).Best.Cost=whales(i).Cost;

        % Update Global Best
        if
whales(i).Best.Cost<GlobalBest.Cost

GlobalBest=whales(i).Best;

        end

        end

        end

    end

    polycheck = 0;
    leader_whales_poly = [];
    temp_whales = [];
    tempeval_whales = [];
    for t=1:nTeam
        for j=1:Dimension
            leader_whales_std(t,j) =
std2(whalesPosition(((t-
1)*ceil(nPop/nTeam))+1:((t-
1)*ceil(nPop/nTeam))+TeamSize(t),j));
            if leader_whales_std < 0.01
                polycheck = 1;
            else
                buffer =
polyfit(whalesPosition(((t-
1)*ceil(nPop/nTeam))+1:((t-

```

```

1)*ceil(nPop/nTeam))+TeamSize(t),j),whalesPos
ition(((t-1)*ceil(nPop/nTeam))+1:(t-
1)*ceil(nPop/nTeam))+TeamSize(t),Dimension+1)
,Porder);

leader_whales_poly(t,:,j) =
buffer;

if max(isnan(buffer)) < 1
    syms x;
    fun =
matlabFunction(poly2sym(buffer));

leader_whales(t).Position(j) =
fminsearch((fun),0);
end
leader_whales(t).Position(j)
= min(leader_whales(t).Position(j), UB(j));
leader_whales(t).Position(j)
= max(leader_whales(t).Position(j), LB(j));
end

end

% temp_whales(:, t) =
leader_whales(t).Position;
end
% tempeval_whales =
CostFunction(temp_whales);
for t=1:nTeam
    % leader_whales(t).Cost =
tempeval_whales(t);
leader_whales(t).Cost =
CostFunction(leader_whales(t).Position);
end

BestCost(it)=GlobalBest.Cost;

disp(['Iteration ' num2str(it) ': Best
Cost = ' num2str(BestCost(it)) ': Datacheck =
' num2str(polycheck)]);

```

```

        w=w*wdamp;
end
BestSol = GlobalBest;

%% Results
min_variable_design = BestSol.Position(1,:)
min_objective_function = BestSol.Cost(1,:)
figure;
%plot(BestCost,'LineWidth',2);
semilogy(BestCost,'LineWidth',2);
title('Grafik Nilai Minimum KWA','color','b')
xlabel('Jumlah Iterasi');
ylabel('Nilai Fungsi Obyektif');
grid on;

```

- *Particle Swarm Optimization*

```

clc;
clear;
close all;

%% Problem Definition

CostFunction=@(x) (model(x));           % Cost
Function

nVar=3;                                % Number of Decision
Variables

VarSize=[1 nVar];    % Size of Decision
Variables Matrix

VarMin=[0.6 0.0127 5];    % Lower Bound of
Variables
VarMax=[0.9 0.02224 7];    % Upper Bound of
Variables

%% PSO Parameters

MaxIt=200;    % Maximum Number of
Iterations

nPop=200;    % Population Size (Swarm
Size)

% PSO Parameters
w=1;    % Inertia Weight
wdamp=0.99;    % Inertia Weight Damping
Ratio
c1=1.5;    % Personal Learning
Coefficient
c2=2.0;    % Global Learning
Coefficient

```



```

% If you would like to use Constriction
Coefficients for PSO,
% uncomment the following block and comment
the above set of parameters.

% % Constriction Coefficients
% phi1=2.05;
% phi2=2.05;
% phi=phi1+phi2;
% chi=2/(phi-2+sqrt(phi^2-4*phi));
% w=chi;           % Inertia Weight
% wdamp=1;         % Inertia Weight Damping
Ratio
% c1=chi*phi1;     % Personal Learning
Coefficient
% c2=chi*phi2;     % Global Learning
Coefficient

% Velocity Limits
VelMax=0.1*(VarMax-VarMin);
VelMin=-VelMax;

%% Initialization

empty_particle.Position=[];
empty_particle.Cost=[];
empty_particle.Velocity=[];
empty_particle.Best.Position=[];
empty_particle.Best.Cost=[];

particle= repmat(empty_particle,nPop,1);

GlobalBest.Cost=inf;

for i=1:nPop

    % Initialize Position

```

```

particle(i).Position=unifrnd(VarMin,VarMax,VarSize);

    % Initialize Velocity
    particle(i).Velocity=zeros(VarSize);

    % Evaluation

particle(i).Cost=CostFunction(particle(i).Position);

    % Update Personal Best

particle(i).Best.Position=particle(i).Position;
    particle(i).Best.Cost=particle(i).Cost;

    % Update Global Best
    if particle(i).Best.Cost<GlobalBest.Cost

        GlobalBest=particle(i).Best;

    end

end

BestCost=zeros(MaxIt,1);

%% PSO Main Loop

for it=1:MaxIt

    for i=1:nPop

        % Update Velocity
        particle(i).Velocity =
w*particle(i).Velocity ...

```

```

+c1*rand(VarSize).*(particle(i).Best.Position-
particle(i).Position) ...

+c2*rand(VarSize).*(GlobalBest.Position-
particle(i).Position);

    % Apply Velocity Limits
    particle(i).Velocity =
max(particle(i).Velocity, VelMin);
    particle(i).Velocity =
min(particle(i).Velocity, VelMax);

    % Update Position
    particle(i).Position =
particle(i).Position + particle(i).Velocity;

    % Velocity Mirror Effect

IsOutside=(particle(i).Position<VarMin |
particle(i).Position>VarMax);
    particle(i).Velocity(IsOutside)=-
particle(i).Velocity(IsOutside);

    % Apply Position Limits
    particle(i).Position =
max(particle(i).Position, VarMin);
    particle(i).Position =
min(particle(i).Position, VarMax);

    % Evaluation
    particle(i).Cost =
CostFunction(particle(i).Position);

    % Update Personal Best
    if
particle(i).Cost<particle(i).Best.Cost

```

```

particle(i).Best.Position=particle(i).Position;

particle(i).Best.Cost=particle(i).Cost;

        % Update Global Best
        if
particle(i).Best.Cost<GlobalBest.Cost

                GlobalBest=particle(i).Best;

        end

    end

end

BestCost(it)=GlobalBest.Cost;

disp(['Iteration ' num2str(it) ': Best
Cost = ' num2str(BestCost(it))]);

w=w*wdamp;

end

BestSol = GlobalBest;

%% Results

figure;
%plot(BestCost,'LineWidth',2);
semilogy(BestCost,'LineWidth',2);
xlabel('Iteration');
ylabel('Best Cost');
grid on;
save ('PSOalone.mat')

```

- *Genetic Algorithm*

```

clear all
close all
%Pembangkitan Populasi dan Parameter
Npop      = 200;
%populasi
Maxit      = 200;                %iterasi
el         = 0.90;                %elatism
Pc         = 0.8;
%probabilitas crossover
Pm         = 0.01;
%probabilitas mutasi
Nvar       = 3;                  %jumlah
variabel desain yang dioptimasi
Nbit       = 20;                  %jumlah
bit
%Constrain

rb         = [0.6 0.0127 5];      %batas
bawah cari constraint x4 x5 DICARI!!!!
ra         = [0.9 0.02224 7];    %batas atas cari
constraint x4 x5 DICARI!!!!

eBangkit   = [];
Individu   = [];
eIndividu  = [];
david      = [];
Dadatfit   = [];
Datfit     = [];
summary    = [];
eDadatfit  = [];
efitnessmax = [];
eIndividuMax = [];

Bangkit = round(rand(Npop,Nbit*Nvar));
popsize = size(Bangkit,1);

for i = 1:Nvar
    batas(i) = ra(i)-rb(i);

```

```

end
for i = 1:Npop
    for j = 1:Nvar
        Desimal(i,j) =
bi2de(Bangkit(i, ((j*Nbit)-(Nbit-
1)): (j*Nbit)), 'left-msb');
        Individu(i,j) =
        (Desimal(i,j)*batas(:,j)-
batas(:,j)+rb(:,j)*(2^Nbit-1))/(2^Nbit-1);
    end
end

Datfit = [];
variabel = [];
for i = 1:size(Individu,1)
    fitness = model(Individu(i,:));
    Datfit = [Datfit;fitness];
    [fitemax,nmax]=max(Datfit);
end

Dadatfit = [];
for generasi=1:Maxit
    disp('GA processing')
    clear command windows
    clear command history
    clear memory

    if generasi > 1
        sort_fit =
sortrows(sort,Nbit*Nvar+1);
        Individu1 = sort_fit(round((1-
el)*Npop+1):Npop,:);
        remain =
sort_fit(round(el*Npop)+1:Npop,:);

        X = Individu1;
        M = size(X,1);

        sumfitness = sum(Datfit);

```

```

for i=1:M
    Prob(i) = Datfit(i)/sumfitness;
end
for i=2:M
    Prob(i) = Prob(i)+Prob(i-1);
end
for i=1:M
    n=rand;
    k=1;
    for j=1:M-1
        if (n>Prob(j))
            k=j+1;
        end
    end
    Xparents(i,:) = X(k,:);
end

%Crossover
[M,d] = size(Xparents);
Xcrossed = Xparents;
for i=1:2:M-1
    c=rand;
    if (c<=Pc)
        p=ceil((d-1)*rand);
        Xcrossed(i,:) =
[Xparents(i,1:p) Xparents(i+1,p+1:d)];
        Xcrossed(i+1,:) =
[Xparents(i+1,1:p) Xparents(i,p+1:d)];
    end
end
if (M/2~=floor(M/2))
    c=rand;
    if (c<=Pc)
        p=ceil((d-1)*rand);
        str=ceil((M-1)*rand);
        Xcrossed(M,:) =
[Xparents(M,1:p) Xparents(str,p+1:d)]; %the
first child is chosen
    end
end

```

```

%Mutasi
[M,d] = size(Xcrossed);
Xnew=Xcrossed;
for i=1:M
    for j=1:d
        p=rand;
        if (p<=Pm)
            Xnew(i,j)=1-
Xcrossed(i,j);
        end
    end
end

disp('New fitness calculation');

Bangkit =
[Xnew(:,1:Nbit*Nvar);remain(:,1:Nbit*Nvar)];
end
eBangkit = [eBangkit; Bangkit];

for i =1:Npop
    for j = 1:Nvar;
        Desimal(i,j) =
bi2de(Bangkit(i, ((j*Nbit)-(Nbit-
1)): (j*Nbit)), 'left-msb');
        Individu(i,j) =
(Desimal(i,j)*batas(:,j)-
batas(:,j)+rb(:,j)*(2^Nbit-1))/(2^Nbit-1);
    end
end

Datfit = [];
for i = 1:Npop
    fitness = model(Individu(i,:));
    Datfit = [Datfit;fitness];
    [fitemax,nmax] = max(Datfit);
end

```



```

Dadatfit = Datfit;
eDadatfit = [eDadatfit;Dadatfit];
eIndividu = [eIndividu;Individu];
[fitnessmax,nmax] = max(eDadatfit);
efitnessmax = [efitnessmax;fitnessmax];
BangkitMax = eBangkit(nmax,:);
IndividuMax = eIndividu(nmax,:);
eIndividuMax =
[eIndividuMax;IndividuMax];
BangkitMaxlast = BangkitMax;
schedmax = BangkitMax;
sort = [Bangkit Dadatfit];
summary = [summary; sort];
david = [david; Dadatfit];

clc
max_variable_design=eIndividuMax(1,:)
max_objective_function=fitness(1,:)
figure(gcf)
title('Grafik Nilai Maksimum
GA','color','b')
xlabel('Jumlah Iterasi')
ylabel('Nilai Fungsi Obyektif')
hold on
plot(efitnessmax, 'DisplayName',
'efitnessmax', 'YDataSource',
'efitnessmax');
hold on
end

```

- *Duelist Algorithm*

```

clear all
clc

Dimension = 3;

UB = [0.9 0.02224 7];           % Upper
Bounds
LB = [0.5 0.0127 5];           % Lower
Bounds
save ('prop.mat')

clear all;
close all;
clc;

%DUELIST ALGORITHM
%DUELIST ALGORITHM
load ('prop.mat')
Hasilmax=[];
fitnessvector = [];
XDueler=[];
convergemax = [];
convergeiter = [];
DFDAfit = [];
xmax = [];
minmax = 'min';                 % 'max' Maximum or
'min' Minimum
Population = 200;               % Total number of
duelists in a population
MaxGeneration = 200;            % Maximum
Generation/Iteration
FightCapabilities = 50;         % Fighting
Capabilities
Champion = 0.1;                 % Champion
Percentage
ProbLearning = 0.8;             % Learning
Probability

```

```

ProbInnovate = 0.1;           % Innovate
Probability
Luckcoeff = 0.01;           % Luck Coefficient
LuckA = 0;                   % First Duelist Luck
Coefficient
LuckB = 0;                   % Second Duelist
Luck Coefficient
Duelist = [];
Duelisttemp1 = [];
Duelisttemp2 = [];
Duelisttemp3 = [];
DuelistInteger = [];
Datafit = [];
Data1fit = [];
DataSort = [];
ElitDuelist = [];
HMI = [];
DataFDAfit = [];
maxall = [];

for rc = 1:Dimension
    RangeB(rc) = UB(rc) - LB(rc);
end

if (strcmp(minmax,'max'))
    mm = 1;
else
    mm = -1;
end

%====Registrasi Duelist=====
Duelist =
floor(9*rand(Population,(FightCapabilities*D
imension))+rand());

%====Array to Int=====

```

```

for i = 1:Dimension
    for j = 1:Population
        Duelisttemp1 =
        Duelist(j, ((i*FightCapabilities-
        FightCapabilities)+1):(i*FightCapabilities))
        ;
        Duelisttemp2 =
        num2str(Duelisttemp1);
        Duelisttemp3 =
        Duelisttemp2(~isspace(Duelisttemp2));
        DuelistInteger(j,i) =
        str2num(Duelisttemp3);
    end
end

```

```
Datafit = [];
```

```

disp('DA Processing');
for Generasi = 1:MaxGeneration

```

```
    %====DA Processing=====
```

```

    if (Generasi > 1)
        clc
        Generasi

```

```

        %====sortir=====
        sort_fit = sortrows(sort,
        (FightCapabilities*Dimension) + 1);
        Duelist1 =
        sort_fit(randperm(size(sort_fit,1)),:);
        Remain = sort_fit(round((1-
        Champion)*Population) + 1:Population, :);
        Winner = [];

```

```

X = Duelist1;
N = size(X,1);

if mod(N,2) == 0
    M=N;
else
    M=N-1;
end

for i=1:M
    fitnessvector(i) =
X(i, (FightCapabilities*Dimension) + 1);
end

fitnessvector = fitnessvector';

%=====Setting Duelist=====
for i=1:M
    XDueler = X;
end

%=====Setting Duel Arena=====

for i=1:2:M-1
    LuckA =
(fitnessvector(i)*(Luckcoeff +
rand*2*Luckcoeff));
    LuckB =
(fitnessvector(i+1)*(Luckcoeff +
rand*2*Luckcoeff));
    if fitnessvector(i)+LuckA <=
fitnessvector(i+1)+LuckB
        Winner(i) = 0;
        Winner(i+1) = 1;
    end
end

```

```

        elseif fitnessvector(i)+LuckA >
fitnessvector(i+1)+LuckB
            Winner(i) = 1;
            Winner(i+1) = 0;
        end
    end

    %====Skill Transfer + Innovate====

    [M,d] = size(XDueler);
    XAftermatch = XDueler;
    for i=1:2:M-1
        if (Winner(i)==1)
            p = ceil(((d/2)-
1)*rand*ProbLearning);
            str = ceil(p+1+(((d/2)-2-
p)*rand*ProbLearning));
            XAftermatch(i,:) =
[XDueler(i,1:p) XDueler(i+1,p+1:str)
XDueler(i,str+1:d)];
            for j=1:d
                p = rand;
                if (p<=ProbInnovate)
                    XAftermatch(i+1,j) =
abs(floor(rand()*9));
                end
            end
        else
            p = ceil(((d/2)-
1)*rand*ProbLearning);
            str = ceil(p+1+(((d/2)-2-
p)*rand*ProbLearning));
            XAftermatch(i+1,:) =
[XDueler(i+1,1:p) XDueler(i,p+1:str)
XDueler(i+1,str+1:d)];
            XAftermatch(i,:) =
XDueler(i,:);
            for j=1:d
                p = rand;

```

```

                                if (p<=ProbInnovate)
                                    XAftermatch(i,j) =
abs(floor(rand()*9));
                                end
                            end
                        end
                    end

Xnew = XAftermatch;

        sort_fitnew = sortrows(Xnew,
(FightCapabilities*Dimension) + 1);
        Duelistnew =
sort_fitnew(round((Champion)*Population)+1:Population,:);
        Duelist =
[Duelistnew(:,1:(FightCapabilities*Dimension
));Remain(:,1:(FightCapabilities*Dimension))
];

    end;
    ElitDuelist = [ElitDuelist; Duelist];

    for i = 1:Dimension
        for j = 1:Population
            Duelisttemp1 =
Duelist(j,((i*FightCapabilities-
FightCapabilities)+1):(i*FightCapabilities))
;
            Duelisttemp2 =
num2str(Duelisttemp1);
            Duelisttemp3 =
Duelisttemp2(~isspace(Duelisttemp2));
            DuelistInteger(j,i) =
str2num(Duelisttemp3);
        end
    end
end

```

```

Datafit = [];

for k = 1:Population

    for ii=1:Dimension
        x(ii,k) =
            (((DuelistInteger(k,ii)+1)/(10^FightCapabili
ties))*RangeB(ii))+LB(ii);
    end

    %           cost = -
    (((X0(1,k).^2)+(X0(2,k).^2)).^0.5).*cos((X0
(1,k))-
(X0(2,k))))).*exp(cos(((X0(1,k)).*(X0(2,k)+5)
)./7));

    fitness = model(x);
    Datafit = [Datafit; fitness];

end

Data1fit = Datafit;
[fitnessmax, nmax] = max(Data1fit);
DataFDAfit = [DataFDAfit;fitnessmax];
DuelistMax = Duelist(nmax,:);
DuelistMaxLast = DuelistMax;
Hasilmax = DuelistMax;
sort = [Duelist Datafit];
maxall = [maxall; sort];
for i = 1:Dimension
    HasilMaxtemp1 =
Hasilmax(1,(((i*FightCapabilities)-
FightCapabilities)+1):(i*FightCapabilities))
;
    HasilMaxtemp2 =
num2str(HasilMaxtemp1);
    HasilMaxtemp3 =
HasilMaxtemp2(~isspace(HasilMaxtemp2));

```



```

        HasilMaxInt(1,i) =
str2num(HasilMaxtemp3);
    end
    HMIIt = [];
    for ij=1:Dimension
        HMIIt = [HMIIt, HasilMaxInt(1,ij)];
    end
    HMI = [HMI; HMIIt];
end

plot(DataFDAfit);
hold on

[fitnessmaxf, nmaxf] = max(DataFDAfit);
for ik=1:Dimension
    X0maxfix(ik) =
    (((HMI(nmaxf,ik)+1)/(10^FightCapabilities))*
    RangeB(ik))+LB(ik);
end

X0maxfix
[fitnessmaxf, nmaxf] = max(DataFDAfit)

convergemax = [convergemax;fitnessmaxf];
convergeiter = [convergeiter;nmaxf];
xmax = [xmax;X0maxfix];
DFDAfit = [DFDAfit,DataFDAfit];

```

Halaman ini sengaja dikosongkan

BIODATA PENULIS



Nama lengkap penulis adalah Anggi Malwindasari, lahir di kota Balikpapan pada tanggal 19 Agustus 1994 dari ayah bernama Sumali dan ibu bernama Winartin. Penulis merupakan anak pertama dari dua bersaudara. Pada tahun 2007 penulis menyelesaikan pendidikan Sekolah Dasar di SD Negeri I Kampungdalem, pada tahun 2010 menyelesaikan pendidikan Sekolah Menengah Pertama di SMP Negeri 1 Tulungagung, pada tahun 2013 menyelesaikan pendidikan Sekolah Menengah Atas di SMA Negeri 1 Boyolangu. Pada tahun yang sama, penulis terdaftar sebagai mahasiswa di Departemen Teknik Fisika Institut Teknologi Sepuluh Nopember.

Penulis telah aktif dalam beberapa organisasi kemahasiswaan diantaranya menjadi pengurus Himpunan Mahasiswa Teknik Fisika ITS, Laboratorium Rekayasa Instrumentasi dan Kontrol, Lembaga Dakwah Jurusan Fusi Ulul Albab dan Paguyuban Karya Salemba Empat ITS. Penulis pernah menjadi salah satu penerima dana PKM DIKTI di bidang karsa cipta pada tahun 2015/2016.

Konsentrasi tugas akhir yang didalami adalah bidang rekayasa instrumentasi dan kontrol. Pada bulan Juli 2017 penulis telah menyelesaikan Tugas Akhir dengan judul **Optimisasi Fouling Resistance pada Shell and Tube Heat Exchanger Menggunakan Polley Model dan Rain Water Optimization.**

Apabila pembaca ingin berdiskusi lebih lanjut mengenai tugas akhir, serta memberikan kritik dan saran maka dapat menghubungi penulis melalui email : malwindasarianggi@gmail.com